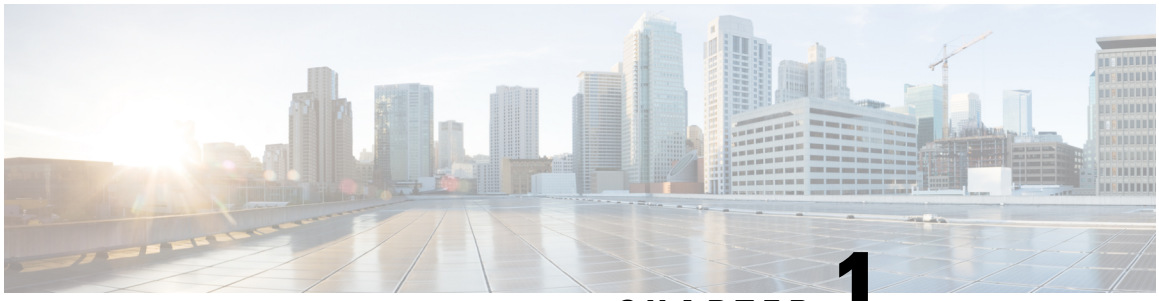
IP Addressing: DHCP Configuration Guide, Cisco IOS XE 17 (Cisco ASR 920 Series)

First Published: 2023-03-31

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

DHCP Overview

The Dynamic Host Configuration Protocol (DHCP) is based on the Bootstrap Protocol (BOOTP), which provides the framework for passing configuration information to hosts on a TCP/IP network. DHCP adds the capability to automatically allocate reusable network addresses and configuration options to Internet hosts. DHCP consists of two components: a protocol for delivering host-specific configuration parameters from a DHCP server to a host and a mechanism for allocating network addresses to hosts. DHCP is built on a client/server model, where designated DHCP server hosts allocate network addresses and deliver configuration parameters to dynamically configured hosts.

This module describes the concepts needed to understand Cisco IOS XE DHCP.

Information About DHCP

DHCP Overview

Cisco routers running Cisco IOS XE software include Dynamic Host Control Protocol (DHCP) server and relay agent software. The Cisco IOS XE DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the router to DHCP clients. These address pools can also be configured to supply additional information to the requesting client such as the IP address of the DNS server, the default router, and other configuration parameters. If the Cisco IOS XE DHCP server cannot satisfy a DHCP request from its own database, it can forward the request to one or more secondary DHCP servers defined by the network administrator.

DHCP supports three mechanisms for IP address allocation:

- Automatic allocation—DHCP assigns a permanent IP address to a client.
- Dynamic allocation—DHCP assigns an IP address to a client for a limited period of time, which is called a lease (or until the client explicitly relinquishes the address). DHCP also supports on-demand address pools (ODAPs), which is a feature in which pools of IP addresses can be dynamically increased or reduced in size depending on the address utilization level. ODAPs support address assignment for customers using private addresses.

- Manual allocation—The network administrator assigns an IP address to a client and DHCP is used simply to convey the assigned address to the client.

The format of DHCP messages is based on the format of BOOTP messages, which ensures support for BOOTP relay agent functionality and interoperability between BOOTP clients and DHCP servers. BOOTP relay agents eliminate the need for deploying a DHCP server on each physical network segment. BOOTP is explained in RFC 951, *Bootstrap Protocol (BOOTP)*, and RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*.

The main advantage of DHCP compared to BOOTP is that DHCP does not require that the DHCP server be configured with all MAC addresses of all clients. DHCP defines a process by which the DHCP server knows the IP subnet in which the DHCP client resides, and it can assign an IP address from a pool of valid IP addresses in that subnet. Most of the other information that DHCP might supply, such as the default router IP address, is the same for all hosts in the subnet so DHCP servers can usually configure information per subnet rather than per host. This functionality reduces network administration tasks compared to BOOTP.

Benefits of Using DHCP

The DHCP implementation offers the following benefits:

- Reduced Internet access costs

  Using automatic IP address assignment at each remote site substantially reduces Internet access costs. Static IP addresses are considerably more expensive to purchase than are automatically allocated IP addresses.

- Reduced server configuration tasks and costs

  Because DHCP is easy to configure, it minimizes operational overhead and costs associated with device configuration tasks and eases deployment by nontechnical users.

- Centralized management

  Because the DHCP server maintains configurations for several subnets, an administrator only needs to update a single, central server when configuration parameters change.

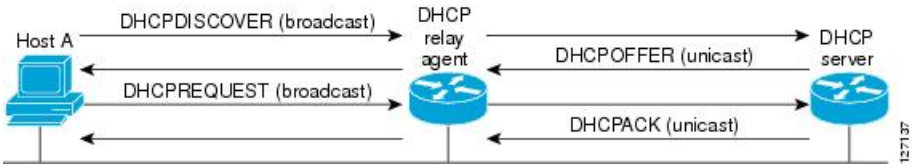
DHCP Server Relay Agent and Client Operation

Dynamic Host Control Protocol (DHCP) provides a framework for passing configuration information dynamically to hosts on a TCP/IP network. A DHCP client is a host that uses DHCP to obtain configuration parameters such as an IP address.

A DHCP relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet. Relay agent forwarding is distinct from the normal forwarding of an IP router, where IP datagrams are switched between networks somewhat transparently. By contrast, relay agents receive DHCP messages and then generate a new DHCP message to send on another interface.

The figure below shows the basic steps that occur when a DHCP client requests an IP address from a DHCP server. The client, Host A, sends a DHCPDISCOVER broadcast message to locate a DHCP server. A relay agent forwards the packets between the DHCP client and server. A DHCP server offers configuration parameters (such as an IP address, a MAC address, a domain name, and a lease for the IP address) to the client in a DHCPOFFER unicast message.

Figure 1: DHCP Request for an IP Address from a DHCP Server



A DHCP client may receive offers from multiple DHCP servers and can accept any one of the offers; however, the client usually accepts the first offer it receives. Additionally, the offer from the DHCP server is not a guarantee that the IP address will be allocated to the client; however, the server usually reserves the address until the client has had a chance to formally request the address.

The client returns a formal request for the offered IP address to the DHCP server in a DHCPREQUEST broadcast message. The DHCP server confirms that the IP address has been allocated to the client by returning a DHCPACK unicast message to the client.

The formal request for the offered IP address (the DHCPREQUEST message) that is sent by the client is broadcast so that all other DHCP servers that received the DHCPDISCOVER broadcast message from the client can reclaim the IP addresses that they offered to the client.

If the configuration parameters sent to the client in the DHCPOFFER unicast message by the DHCP server are invalid (a misconfiguration error exists), the client returns a DHCPDECLINE broadcast message to the DHCP server.

The DHCP server will send to the client a DHCPNAK denial broadcast message, which means the offered configuration parameters have not been assigned, if an error has occurred during the negotiation of the parameters or the client has been slow in responding to the DHCPOFFER message (the DHCP server assigned the parameters to another client) of the DHCP server.

DHCP Database

DHCP address pools are stored in non-volatile RAM (NVRAM). There is no limit on the number of address pools. An address binding is the mapping between the client's IP and hardware addresses. The client's IP address can be configured by the administrator (manual address allocation) or assigned from a pool by the DHCP server.

Manual bindings are stored in NVRAM. Manual bindings are just special address pools configured by a network administrator. There is no limit on the number of manual bindings.

Automatic bindings are IP addresses that have been automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Automatic bindings are stored on a remote host called the database agent. A DHCP database agent is any host--for example, an FTP, TFTP, or RCP server--that stores the DHCP bindings database.The bindings are saved as text records for easy maintenance.

You can configure multiple DHCP database agents and you can configure the interval between database updates and transfers for each agent.

DHCP Attribute Inheritance

The DHCP server database is organized as a tree. The root of the tree is the address pool for natural networks, branches are subnetwork address pools, and leaves are manual bindings to clients. Subnetworks inherit network parameters and clients inherit subnetwork parameters. Therefore, common parameters (for example, the domain name) should be configured at the highest (network or subnetwork) level of the tree.

Inherited parameters can be overridden. For example, if a parameter is defined in both the natural network and a subnetwork, the definition of the subnetwork is used.

Address leases are not inherited. If a lease is not specified for an IP address, by default, the DHCP server assigns a one-day lease for the address.

DHCP Options and Suboptions

Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. Options provide a method of appending additional information. Vendors that want to provide additional information to their client not designed into the protocol can use options.

The Cisco IOS XE DHCP implementation also allows most DHCP server options to be customized. For example, the TFTP server, which stores the Cisco IOS XE image, can be customized with option 150 to support intelligent IP phones.

Virtual Private Networks (VPNs) allow the possibility that two pools in separate networks can have the same address space, with private network addresses, served by the same DHCP server. Cisco IOS XE software supports VPN-related options and suboptions such as the relay agent information option and VPN identification suboption. A relay agent can recognize these VPN-related options and suboptions and forward the client-originated DHCP packets to a DHCP server. The DHCP server can use this information to assign IP addresses and other parameters, distinguished by a VPN identifier, to help select the VPN to which the client belongs.

For more information on DHCP options and suboptions, see the “DHCP Options Reference” appendix in the *Network Registrar User’s Guide* , Release 6.3.

During lease negotiation, the DHCP server sends the options shown in the table below to the client.

Table 1: Default DHCP Server Options

DHCP Option Name	DHCP Option Code	Description
Subnet mask option	1	Specifies the client’s subnet mask per RFC 950.
Router option	3	Specifies a list of IP addresses for routers on the client’s subnet, usually listed in order of preference.
Domain name server option	6	Specifies a list of DNS name servers available to the client, usually listed in order of preference.
Hostname option	12	Specifies the name of the client. The name may or may not be qualified with the local domain name.
Domain name option	15	Specifies the domain name that the client should use when resolving hostnames via the Domain Name System.
NetBIOS over TCP/IP name server option	44	Specifies a list of RFC 1001/1002 NetBIOS name servers listed in order or preference.
NetBIOS over TCP/IP node type option	46	Enables NetBIOS over TCP/IP clients that are configurable to be configured as described in RFC 1001/1002.
IP address lease time option	51	Allows the client to request a lease for the IP address.

DHCP Option Name	DHCP Option Code	Description
DHCP message type option	53	Conveys the type of the DHCP message.
Server identifier option	54	Identifies the IP address of the selected DHCP server.
Renewal (T1) time option	58	Specifies the time interval from address assignment until the client transitions to the renewing state.
Rebinding (T2) time option	59	Specifies the time interval from address assignment until the client transitions to the rebinding state.

DHCP Server On-Demand Address Pool Management Overview

The Cisco IOS DHCP server on-demand address pool (ODAP) manager is used to centralize the management of large pools of addresses and simplify the configuration of large networks. ODAP provides a central management point for the allocation and assignment of IP addresses. When a router is configured as an ODAP manager, pools of IP addresses are dynamically increased or reduced in size depending on the address utilization level.

ODAPs support address assignment using DHCP for customers using private addresses. Each ODAP is configured and associated with a particular Multiprotocol Label Switching (MPLS) VPN. Cisco IOS software also provides ODAP support for non-MPLS VPN address pools by adding pool name support to the **peer default ip address dhcp-pool** *pool name*command.

DHCP server subnet allocation is a way of offering entire subnets (ranges of addresses) to relay agents so that remote access devices can provision IP addresses to DHCP clients. This functionality can occur along with or instead of managing individual client addresses. Subnet allocation can improve IP address provisioning, aggregation, characterization, and distribution by relying on the DHCP infrastructure to dynamically manage subnets.

This capability allows the DHCP server to be configured with a pool of subnets for lease to ODAP clients. Subnet pools can be configured for global ODAP clients or MPLS VPN ODAP clients on a per-client basis. The DHCP subnet allocation server creates bindings for the subnet leases and stores these leases in the DHCP database.

Additional References for DHCP Overview

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
DHCP commands	Cisco IOS IP Addressing Services Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFC 951	*Bootstrap Protocol (BOOTP)*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	https://www.cisco.com/c/en/us/support/index.html

Glossary

address binding—A mapping between the client's IP and hardware (MAC) addresses. The client's IP address may be configured by the administrator (manual address allocation) or assigned from a pool by the DHCP server (automatic address allocation). The binding also contains a lease expiration date. The default for the lease expiration date is one day.

address conflict—A duplication of use of the same IP address by two hosts. During address assignment, DHCP checks for conflicts using ping and gratuitous (ARP). If a conflict is detected, the address is removed from the pool. The address will not be assigned until the administrator resolves the conflict.

address pool—The range of IP addresses assigned by the DHCP server. Address pools are indexed by subnet number.

automatic address allocation --An address assignment method where a network administrator obtains an IP address for a client for a finite period of time or until the client explicitly relinquishes the address. Automatic allocation is particularly useful for assigning an address to a client that will be connected to the network only temporarily or for sharing a limited pool of IP addresses among a group of clients that do not need permanent IP addresses. Automatic allocation may also be a good choice for assigning an IP address to a new client being permanently connected to a network where IP addresses are sufficiently scarce that it is important to reclaim them when old clients are retired.

BOOTP—Bootstrap Protocol. A protocol that provides a method for a booting computer to find out its IP address and the location of the boot file with the rest of its parameters.

client—Any host requesting configuration parameters.

database—A collection of address pools and bindings.

database agent—Any host storing the DHCP bindings database, for example, a Trivial File Transfer Protocol (TFTP) server.

DHCP—Dynamic Host Configuration Protocol. A protocol that provides a mechanism for allocating IP addresses dynamically so that addresses can be reused when hosts no longer need them.

DNS—Domain Name System. A system used in the Internet for translating names of network nodes into addresses.

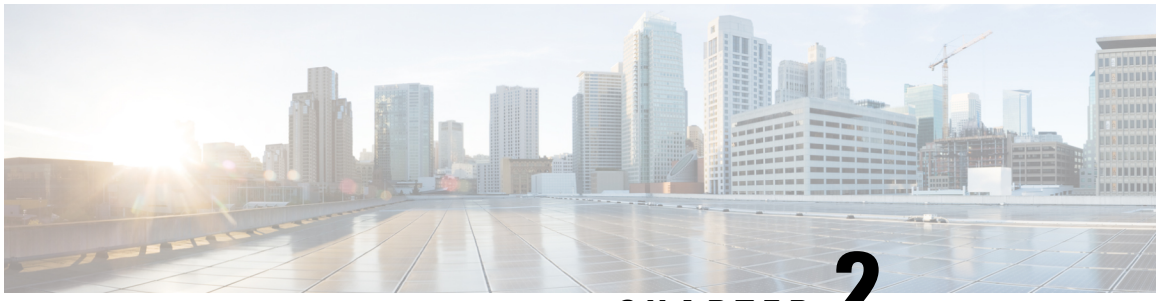
manual address allocation—An address assignment method that allocates an administratively assigned IP address to a host. Manual allocation allows DHCP to be used to eliminate the error-prone process of manually configuring hosts with IP addresses.

PWLAN—Public Wireless Local Area Network. A type of wireless LAN, often referred to as a hotspot, that anyone having a properly configured computer device can access.

relay agent—A device that forwards DHCP and BOOTP messages between a server and a client on different subnets.

server—Any host providing configuration parameters.

SSG—Service Selection Gateway. The feature set that provides on-demand service enforcement within the Cisco network.



CHAPTER 2

Configuring the Cisco IOS XE DHCP Server

Cisco routers running Cisco IOS XE software include Dynamic Host Configuration Protocol (DHCP) server and relay agent software. The Cisco IOS XE DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the router to DHCP clients. The DHCP server can be configured to assign additional parameters such as the IP address of the domain name system (DNS) server and the default router.

This module describes the concepts and the tasks needed to configure the DHCP server.

Prerequisites for Configuring the DHCP Server

- Before you configure a Cisco Dynamic Host Control Protocol (DHCP) server, you must understand the concepts documented in the Overview of the DHCP Server section.
- The Cisco DHCP server and the relay agent services are enabled by default. Use the **no service dhcp** command to disable the Cisco DHCP server and the relay agent and the **service dhcp** command to reenable the functionality.
- Port 67 (the DHCP server port) is closed in the Cisco DHCP/BOOTP default configuration. There are two logical parts to the **service dhcp** command: service enabled and service running. The DHCP service is enabled by default, but port 67 does not open until the DHCP service is running. If the DHCP service is running, the **show ip sockets details** or the **show sockets detail** command displays port 67 as open.
- The Cisco DHCP relay agent is enabled on an interface only when you configure the **ip helper-address** command. This command enables a DHCP broadcast to be forwarded to the configured DHCP server.

Information About the Cisco IOS XE DHCP Server

Overview of the DHCP Server

The Cisco DHCP server accepts address assignment requests and renewals from the client and assigns the addresses from predefined groups of addresses within DHCP address pools. These address pools can also be configured to supply additional information to the requesting client such as the IP address of the Domain Name System (DNS) server, the default device, and other configuration parameters. The Cisco DHCP server can accept broadcasts from locally attached LAN segments or from DHCP requests that have been forwarded by other DHCP relay agents within the network.

Database Agents

A DHCP database agent is any host (for example, an FTP, TFTP, or rcp server) or storage media on the DHCP server (for example, flash disk) that stores the DHCP bindings database. You can configure multiple DHCP database agents, and you can configure the interval between database updates and transfers for each agent.

Automatic bindings are IP addresses that have been automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Automatic binding information (such as lease expiration date and time, interface index, and VPN routing and forwarding [VRF] name) is stored on a database agent. The bindings are saved as text records for easy maintenance.

Address Conflicts

An address conflict occurs when two hosts use the same IP address. During address assignment, DHCP checks for conflicts using ping and gratuitous Address Resolution Protocol (ARP). If a conflict is detected, the address is removed from the pool. The address will not be assigned until the administrator resolves the conflict.

DHCP Address Pool Conventions

You can configure a DHCP address pool with a name that is a symbolic string (such as “engineering”) or an integer (such as 0). Configuring a DHCP address pool also puts the router into DHCP pool configuration mode--identified by the (dhcp-config)# prompt--from which you can configure pool parameters (for example, the IP subnet number and default router list).

DHCP Address Pool Selection

DHCP defines a process by which the DHCP server knows the IP subnet in which the DHCP client resides, and it can assign an IP address from a pool of valid IP addresses in that subnet. The process by which the DHCP server identifies which DHCP address pool to use to service a client request is described in this section.

The DHCP server identifies which DHCP address pool to use to service a client request as follows:

- If the client is not directly connected (the giaddr field of the DHCPDISCOVER broadcast message is non-zero), the DHCP server matches the DHCPDISCOVER with a DHCP pool that has the subnet that contains the IP address in the giaddr field. Giaddr field is the gateway IP address field of a DHCP packet. A DHCP relay agent sets the gateway address and adds the relay agent information option (option82) in the packet and forwards it to the DHCP server. The reply from the server is forwarded back to the client after removing option 82.

- If the client is directly connected (the giaddr field is zero), the DHCP server matches the DHCPDISCOVER with DHCP pool(s) that contain the subnet(s) configured on the receiving interface. If the interface has secondary IP addresses, the subnets associated with the secondary IP addresses are examined for possible allocation only after the subnet associated with the primary IP address (on the interface) is exhausted.

Cisco IOS XE DHCP server software supports advanced capabilities for IP address allocation. See the "DHCP Server Address Allocation Using Option 82" section for more information.

Address Bindings

An address binding is a mapping between the IP address and MAC address of a client. The IP address of a client can be assigned manually by an administrator or assigned automatically from a pool by a DHCP server.

Manual bindings are IP addresses that have been manually mapped to the MAC addresses of hosts that are found in the DHCP database. Manual bindings are stored in NVRAM on the DHCP server. Manual bindings are just special address pools. There is no limit on the number of manual bindings, but you can only configure one manual binding per host pool.

Automatic bindings are IP addresses that have been automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Because the bindings are stored in volatile memory on the DHCP server, binding information is lost in the event of a power failure or upon router reload for any other reason. To prevent the loss of automatic binding information in such an event, a copy of the automatic binding information can be stored on a remote host called a DHCP database agent. The bindings are periodically written to the database agent. If the router reloads, the bindings are read back from the database agent to the DHCP database on the DHCP server.

Note We strongly recommend using database agents. However, the Cisco IOS XE DHCP server can function without database agents.

All DHCP clients send a client identifier (DHCP option 61) in the DHCP packet. To configure manual bindings, you must enter the **client-identifier** DHCP pool configuration command with the appropriate hexadecimal values identifying the DHCP client.

Ping Packet Settings

By default, the DHCP server pings a pool address twice before assigning a particular address to a requesting client. If the ping is unanswered, the DHCP server assumes (with a high probability) that the address is not in use and assigns the address to the requesting client.

By default, the DHCP server waits 2 seconds before timing out a ping packet.

DHCP Attribute Inheritance

The DHCP server database is organized as a tree. The root of the tree is the address pool for natural networks, branches are subnetwork address pools, and leaves are manual bindings to clients. Subnetworks inherit network parameters and clients inherit subnetwork parameters. Therefore, common parameters (for example, the domain name) should be configured at the highest (network or subnetwork) level of the tree.

Inherited parameters can be overridden. For example, if a parameter is defined in both the natural network and a subnetwork, the definition of the subnetwork is used.

Address leases are not inherited. If a lease is not specified for an IP address, by default, the DHCP server assigns a one-day lease for the address.

DHCP Server Address Allocation Using Option 82

The Cisco IOS DHCP server can allocate dynamic IP addresses based on the relay information option (option 82) sent by the relay agent.

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The data items are also called options. Option 82 is organized as a single DHCP option that contains information known by the relay agent.

Automatic DHCP address allocation is based on an IP address. This IP address can either be the gateway address (giaddr field of the DHCP packet) or the IP address of an incoming interface. In some networks, it is necessary to use additional information to further determine which IP addresses to allocate. By using option 82, the Cisco IOS DHCP relay agent can include additional information about itself when forwarding client-originated DHCP packets to a DHCP server. The Cisco IOS DHCP server can also use option 82 to provide additional information to properly allocate IP addresses to DHCP clients. The information sent via option 82 is used to identify the port where the DHCP request arrives. Automatic DHCP address allocation does not parse out the individual suboptions contained in option 82. Rather, the address allocation is done by matching a configured pattern byte by byte.

This feature introduces a new DHCP class capability, which is a method to group DHCP clients based on some shared characteristics other than the subnet in which the clients reside.

For example, DHCP clients are connected to two ports of a single switch. Each port can be configured to be a part of two VLANs: VLAN1 and VLAN2. DHCP clients belong to either VLAN1 or VLAN2 and the switch can differentiate the VLAN that a particular DHCP Discover message belongs to (possibly through Layer 2 encapsulation). Each VLAN has its own subnet and all DHCP messages from the same VLAN (same switch) have the giaddr field set to the same value indicating the subnet of the VLAN.

Problems can occur while allocating IP addresses to DHCP clients that are connected to different ports of the same VLAN. These IP addresses must be part of the same subnet but the range of IP addresses must be different. In the preceding example, when a DHCP client that is connected to a port of VLAN1 must be allocated an IP address from a range of IP addresses within the VLAN's subnet, whereas a DHCP client connecting to port 2 of VLAN1 must be allocated an IP address from another range of IP addresses. The two range of IP addresses are part of the same subnet (and have the same subnet mask). Generally, during DHCP address allocation, the DHCP server refers only to the giaddr field and is unable to differentiate between the two ranges.

To solve this problem, a relay agent residing at the switch inserts the relay information option (option 82), which carries information specific to the port, and the DHCP server inspects both the giaddr field and the inserted option 82 during the address selection process.

When you enable option 82 on a device, the following sequence of events occurs:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.
2. When the device receives the DHCP request, it adds the option 82 information in the packet. The option 82 information contains the device MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption).
3. The device adds the IP address of the relay agent to the DHCP packet.
4. The device forwards the DHCP request that includes the option 82 field to the DHCP server.

5. The DHCP server receives the packet. If the server is option 82 capable, it uses the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server echoes the option 82 field in the DHCP reply.
6. The DHCP server unicasts the reply to the device if the request is relayed to the server by the device. The device verifies that it originally inserted the option 82 data by inspecting remote ID and possibly circuit ID fields. The device removes the option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

The Cisco software refers to a pool of IP addresses (giaddr or incoming interface IP address) and matches the request to a class or classes configured in the pool in the order the classes are specified in the DHCP pool configuration.

When a DHCP address pool is configured with one or more DHCP classes, the pool becomes a restricted access pool, which means that no addresses are allocated from the pool unless one or more classes in the pool matches. This design allows DHCP classes to be used either for access control (no default class is configured on the pool) or to provide further address range partitions within the subnet of the pool.

Multiple pools can be configured with the same class, eliminating the need to configure the same pattern in multiple pools.

The following capabilities are supported for DHCP class-based address allocation:

- Specifying the full relay agent information option value as a raw hexadecimal string by using the **relay-information hex** command in new relay agent information configuration mode.
- Support for bit-masking the raw relay information hexadecimal value.
- Support for a wildcard at the end of a hexadecimal string specified by the **relay-information hex** command.

If the relay agent inserts option 82 but does not set the giaddr field in the DHCP packet, the DHCP server interface must be configured as a trusted interface by using the **ip dhcp relay information trusted** command. This configuration prevents the server from dropping the DHCP message.

DHCP Address Allocation Using Option 82 Feature Design

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The data items themselves are also called options. Option 82 is organized as a single DHCP option that contains information known by the relay agent.

This feature is designed to allow the Cisco IOS XE DHCP server to use option 82 information to help determine which IP addresses to allocate to clients. The information sent via option 82 will be used to identify which port the DHCP request came in on. This feature does not parse out the individual suboptions contained within option 82. Rather, the address allocation is done by matching a configured pattern byte by byte.

The feature introduces a new DHCP class capability, which is a method to group DHCP clients based on some shared characteristics other than the subnet in which the clients reside.

Usage Scenario for DHCP Address Allocation Using Option 82

In an example application, DHCP clients are connected to two ports of a single switch. Each port can be configured to be part of two VLANs: VLAN1 and VLAN2. DHCP clients belong to either VLAN1 or VLAN2 and it is assumed that the switch can differentiate the VLAN that a particular DHCP Discover message belongs to (possibly through Layer 2 encapsulation). Each VLAN has its own subnet and all DHCP messages from

the same VLAN (same switch) will have the giaddr field set to the same value indicating the subnet of the VLAN.

The problem is that for a DHCP client connecting to port 1 of VLAN1, it must be allocated an IP address from one range within the VLAN's subnet, whereas a DHCP client connecting to port 2 of VLAN1 must be allocated an IP address from another range. Both these two IP address ranges are part of the same subnet (and have the same subnet mask). In the normal DHCP address allocation, the DHCP server will look only at the giaddr field and thus will not be able to differentiate between the two ranges.

To solve this problem, a relay agent residing at the switch inserts the relay information option (option 82), which carries information specific to the port, and the DHCP server must inspect both the giaddr field and the inserted option 82 during the address selection process.

DHCP Class Capability

The Cisco IOS XE software will look up a pool based on IP address (giaddr or incoming interface IP address) and then match the request to a class or classes configured in the pool in the order the classes are specified in the DHCP pool configuration.

When a DHCP address pool has been configured with one or more DHCP classes, the pool becomes a restricted access pool, which means that no addresses will be allocated from the pool unless one or more of the classes in the pool is matched. This design allows DHCP classes to be used for either access control (no default class is configured on the pool) or to provide further address range partitions with the subnet of the pool.

Multiple pools can be configured with the same class, eliminating the need to configure the same pattern in multiple pools.

The following capabilities are currently supported for DHCP class-based address allocation:

- Specifying the full relay agent information option value as a raw hexadecimal string by using the **relay-information hex** command in the new relay agent information configuration mode.
- Support for bitmasking the raw relay information hexadecimal value.
- Support for a wildcard at the end of the hexadecimal string specified by the **relay-information hex** command.

RegEx and Longest Match Support

DHCP server software supports advanced capabilities for IP address allocation. Earlier, DHCP server supported only exact match on hexadecimal codes. Effective with Cisco IOS XE Fuji 16.9.1, DHCP server is enhanced to support Regular expression (RegEx) based match or longest match. DHCP server provides options to set of DHCP clients with Vendor Class ID (VCI). Each set of clients are serviced from specific DHCP pool with one or more Vendor Classes. RegEx based Vendor Class Identifier match is included to support this feature.

For one class option, either Exact Match or Regex Match or Longest Match is supported. The configured Regex or hexadecimal string is matched against VCI string received in DHCP packets. In case of successful match, server assigns an IP address from the address range specified in pool class configuration. In case of multiple class match, the first occurrence of the match is considered. In case of no match, no address is allocated.

How to Configure the Cisco IOS XE DHCP Server

Configuring a DHCP Database Agent or Disabling Conflict Logging

A DHCP database agent is any host (for example, an FTP, a TFTP, or a remote copy protocol [RCP] server) or storage media on a DHCP server (for example, disk0) that stores the DHCP bindings database. You can configure multiple DHCP database agents, and the interval between database updates and transfers for each agent.

Automatic bindings are IP addresses that are automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Automatic binding information (such as lease expiration date and time, interface index, and VPN routing and forwarding [VRF] name) is stored in a database agent. The bindings are saved as text records for easy maintenance.

An address conflict occurs when two hosts use the same IP address. During address assignment, DHCP checks for conflicts by using ping and gratuitous Address Resolution Protocol (ARP). If a conflict is detected, the address is removed from the pool. The address is not assigned until the administrator resolves the conflict.

Note We strongly recommend using database agents. However, the Cisco DHCP server can run without database agents. If you choose not to configure a DHCP database agent, disable the recording of DHCP address conflicts on the DHCP server by using the **no ip dhcp conflict logging** command in global configuration mode. If there is a conflict logging but no database agent is configured, bindings during a switchover are lost when a device reboots. Possible false conflicts can occur causing the address to be removed from the address pool.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ip dhcp database** *url* [**timeout** *seconds* | **write-delay** *seconds*]
   - **no ip dhcp conflict logging**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose

Step 3	Do one of the following:
	• **ip dhcp database** *url* [**timeout** *seconds* | **write-delay** *seconds*]
	• **no ip dhcp conflict logging**
	Example:
	`Device(config)# ip dhcp database ftp://user:password@172.16.1.1/router-dhcp timeout 80`
	Example:
	`Device(config)# no ip dhcp conflict logging`
		Configures a DHCP server to save automatic bindings on a remote host called a database agent.
		or
		Disables DHCP address conflict logging.

Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Excluding IP Addresses

The IP address configured on a device interface is automatically excluded from the DHCP address pool. The DHCP server assumes that all other IP addresses in a DHCP address pool subnet are available for assigning to DHCP clients.

You must exclude addresses from the pool if the DHCP server does not allocate those IP addresses to DHCP clients. Consider a scenario where two DHCP servers are set up for the same network segment (subnet) for redundancy. If DHCP servers do not coordinate their services with each other using a protocol such as DHCP failover, each DHCP server must be configured to allocate addresses from a nonoverlapping set of addresses in the shared subnet. See the Configuring Manual Bindings section for a configuration example.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp excluded-address** *low-address* [*high-address*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp excluded-address** *low-address* [*high-address*] **Example:** `Device(config)# ip dhcp excluded-address 172.16.1.100 172.16.1.103`	Specifies IP addresses that the DHCP server should not assign to DHCP clients.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring DHCP Address Pools

Configuring a DHCP Address Pool

On a per-address pool basis, specify DHCP options for the client as necessary.

You can configure a DHCP address pool with a name that is a string (such as “engineering”) or an integer (such as 0). Configuring a DHCP address pool also puts the device into DHCP pool configuration mode—identified by the (dhcp-config)# prompt—from which you can configure pool parameters (for example, the IP subnet number and default device list).

DHCP defines a process by which the DHCP server knows the IP subnet in which the DHCP client resides, and it can assign an IP address from a pool of valid IP addresses in that subnet. The process by which the DHCP server identifies the DHCP address pool to use for a client request is described in the Configuring Manual Bindings section.

The DHCP server identifies and uses DHCP address pools for a client request, in the following manner:

- If the client is not directly connected to the DHCP server (the giaddr field of the DHCPDISCOVER broadcast message is nonzero), the server matches the DHCPDISCOVER with the DHCP pool that has the subnet that contains the IP address in the giaddr field.
- If the client is directly connected to the DHCP server (the giaddr field is zero), the DHCP server matches the DHCPDISCOVER with DHCP pools that contain the subnets configured on the receiving interface. If the interface has secondary IP addresses, subnets associated with the secondary IP addresses are examined for possible allocation only after the subnet associated with the primary IP address (on the interface) is exhausted.
- If you want to unconfigure the last DHCP pool when the DHCP traffic is active, ensure to have one of the following configurations:
  1. **ip dhcp pool** <*dummy_pool*> any dummy pool without any configuration

     OR
  2. **ip helper-address** x.x.x.x on dummy interface(loopback)

Cisco DHCP server software supports advanced capabilities for IP address allocation. See the Configuring DHCP Address Allocation Using Option 82 section for more information.

Before you begin

Before you configure the DHCP address pool, you must:

- Identify DHCP options for devices where necessary, including the following:
  - Default boot image name
  - Default devices
  - Domain Name System (DNS) servers
  - Network Basic Input/Output System (NetBIOS) name server
  - Primary subnet
  - Secondary subnets and subnet-specific default device lists (see Configuring a DHCP Address Pool with Secondary Subnets for information on secondary subnets).
- Decide on a NetBIOS node type (b, p, m, or h).
- Decide on a DNS domain name.

Note You cannot configure manual bindings within the same pool that is configured with the **network** DHCP pool configuration command. To configure manual bindings, see the Configuring Manual Bindings section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **utilization mark high** *percentage-number* [**log**]
5. **utilization mark low** *percentage-number* [**log**]
6. **network** *network-number* [*mask* | */prefix-length*] [**secondary**]
7. **domain-name** *domain*
8. **dns-server** *address* [*address2 ... address8*]
9. **bootfile** *filename*
10. **next-server** *address* [*address2 ... address8*]
11. **netbios-name-server** *address* [*address2 ... address8*]
12. **netbios-node-type** *type*
13. **default-router** *address* [*address2 ... address8*]
14. **option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*}
15. **import** {**all** | **interface** *interface_name*
16. **lease** {*days* [*hours* [*minutes*]] | **infinite**}
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Device(config)# ip dhcp pool 1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.
Step 4	**utilization mark high** *percentage-number* [**log**] **Example:** `Device(dhcp-config)# utilization mark high 80 log`	(Optional) Configures the high utilization mark of the current address pool size. • The **log** keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization exceeds the configured high utilization threshold.
Step 5	**utilization mark low** *percentage-number* [**log**] **Example:** `Device(dhcp-config)# utilization mark low 70 log`	(Optional) Configures the low utilization mark of the current address pool size. • The **log** keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization falls below the configured low utilization threshold.
Step 6	**network** *network-number* [*mask* | */prefix-length*] [**secondary**] **Example:** `Device(dhcp-config)# network 172.16.0.0 /16`	Specifies the subnet network number and mask of the DHCP address pool.
Step 7	**domain-name** *domain* **Example:** `Device(dhcp-config)# domain-name cisco.com`	Specifies the domain name for the client.
Step 8	**dns-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# dns server 172.16.1.103 172.16.2.103`	Specifies the IP address of a DNS server that is available to a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command.

	Command or Action	Purpose
		• Servers should be listed in order of preference.
Step 9	**bootfile** *filename* **Example:** `Device(dhcp-config)# bootfile xllboot`	(Optional) Specifies the name of the default boot image for a DHCP client. • The boot file is used to store the boot image for the client. The boot image is generally the operating system that the client uses to load.
Step 10	**next-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# next-server 172.17.1.103 172.17.2.103`	(Optional) Configures the next server in the boot process of a DHCP client. • One address is required; however, you can specify up to eight addresses in one command line. • If multiple servers are specified, DHCP assigns them to clients in a round-robin order. The first client gets address 1, the next client gets address 2, and so on. • If this command is not configured, DHCP uses the server specified by the **ip helper address** command as the boot server.
Step 11	**netbios-name-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# netbios-name-server 172.16.1.103 172.16.2.103`	(Optional) Specifies the NetBIOS WINS server that is available to a Microsoft DHCP client. • One address is required; however, you can specify up to eight addresses in one command line. • Servers should be listed in order of preference.
Step 12	**netbios-node-type** *type* **Example:** `Device(dhcp-config)# netbios-node-type h-node`	(Optional) Specifies the NetBIOS node type for a Microsoft DHCP client.
Step 13	**default-router** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# default-router 172.16.1.100 172.16.1.101`	(Optional) Specifies the IP address of the default device for a DHCP client. • The IP address should be on the same subnet as the client. • One IP address is required; however, you can specify up to eight IP addresses in one command line. These default devices are listed in order of preference; that is, *address* is the most preferred device, *address2* is the next most preferred device, and so on. • When a DHCP client requests an IP address, the device—acting as a DHCP server—accesses the default device list to select another device that the DHCP client will use as the first hop for forwarding

	Command or Action	Purpose
		messages. After a DHCP client has booted, the client begins sending packets to its default device.
Step 14	**option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*} **Example:** `Device(dhcp-config)# option 19 hex 01`	(Optional) Configures DHCP server options. Configuration supports Longest match and RegEx match for option 60. The **option code** sub command can be used to configure any DHCP options.

Step 15	**import** {**all** | **interface** *interface_name*
	Example:
	`Device(dhcp-config)# import all`
	`Device(dhcp-config) # import interface Ethernet0/0`
		The **import all** command learns options from all the interfaces.
		The **import interface** learns options only from the specified interface.

Step 16	**lease** {*days* [*hours* [*minutes*]] | **infinite**} **Example:** `Device(dhcp-config)# lease 30`	(Optional) Specifies the duration of the lease. • The default is a one-day lease. • The **infinite** keyword specifies that the duration of the lease is unlimited.
Step 17	**end** **Example:** `Device(dhcp-config)# end`	Returns to privileged EXEC mode.

Configuring a DHCP Address Pool with Secondary Subnets

For any DHCP pool, you can configure a primary subnet and any number of secondary subnets. Each subnet is a range of IP addresses that the device uses to allocate an IP address to a DHCP client. The DHCP server multiple subnet functionality enables a Cisco DHCP server address pool to manage additional IP addresses by adding the addresses to a secondary subnet of an existing DHCP address pool (instead of using a separate address pool).

Configuring a secondary DHCP subnetwork places the device in DHCP pool secondary subnet configuration mode—identified by the (config-dhcp-subnet-secondary)# prompt—where you can configure a default address list that is specific to the secondary subnet. You can also specify the utilization rate of the secondary subnet, which allows pools of IP addresses to dynamically increase or reduce in size depending on the address utilization level. This setting overrides the global utilization rate.

If the DHCP server selects an address pool that contains multiple subnets, the DHCP server allocates an IP address from the subnets as follows:

- When the DHCP server receives an address assignment request, it looks for an available IP address in the primary subnet.
- When the primary subnet is exhausted, the DHCP server automatically looks for an available IP address in any of the secondary subnets maintained by the DHCP server (even though the giaddr does not

necessarily match the secondary subnet). The server inspects the subnets for address availability in the order of subnets that were added to the pool.

- If the giaddr matches a secondary subnet in the pool, the DHCP server allocates an IP address from that particular secondary subnet (even if IP addresses are available in the primary subnet and irrespective of the order of secondary subnets that were added).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **utilization mark high** *percentage-number* [**log**]
5. **utilization mark low** *percentage-number* [**log**]
6. **network** *network-number* [*mask* | */prefix-length*]
7. **domain-name** *domain*
8. **dns-server** *address* [*address2 ... address8*]
9. **bootfile** *filename*
10. **next-server** *address* [*address2 ... address8*]
11. **netbios-name-server** *address* [*address2 ... address8*]
12. **netbios-node-type** *type*
13. **default-router** *address* [*address2 ... address8*]
14. **option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*}
15. **lease** {*days* [*hours*] [*minutes*] | **infinite**}
16. **network** *network-number* [*mask* | */prefix-length*] [**secondary**]
17. **override default-router** *address* [*address2 ... address8*]
18. **override utilization high** *percentage-number*
19. **override utilization low** *percentage-number*
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Device(config)# ip dhcp pool 1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.

	Command or Action	Purpose
Step 4	**utilization mark high** *percentage-number* [**log**] **Example:** `Device(dhcp-config)# utilization mark high 80 log`	(Optional) Configures the high utilization mark of the current address pool size. • The **log** keyword enables logging of a system message. A system message is generated for a DHCP pool when the pool utilization exceeds the configured high utilization threshold.
Step 5	**utilization mark low** *percentage-number* [**log**] **Example:** `Device(dhcp-config)# utilization mark low 70 log`	(Optional) Configures the low utilization mark of the current address pool size. • The **log** keyword enables logging of a system message. A system message is generated for a DHCP pool when the pool utilization falls below the configured low utilization threshold.
Step 6	**network** *network-number* [*mask* | */prefix-length*] **Example:** `Device(dhcp-config)# network 172.16.0.0 /16`	Specifies the subnet network number and mask of the primary DHCP address pool.
Step 7	**domain-name** *domain* **Example:** `Device(dhcp-config)# domain-name cisco.com`	Specifies the domain name for the client.
Step 8	**dns-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# dns server 172.16.1.103 172.16.2.103`	Specifies the IP address of a DNS server that is available to a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command. • Servers should be listed in the order of preference.
Step 9	**bootfile** *filename* **Example:** `Device(dhcp-config)# bootfile xllboot`	(Optional) Specifies the name of the default boot image for a DHCP client. • The boot file is used to store the boot image for the client. The boot image is generally the operating system image that the client loads.
Step 10	**next-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# next-server 172.17.1.103 172.17.2.103`	(Optional) Configures the next server in the boot process of a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command line. • If multiple servers are specified, DHCP assigns the servers to clients in a round-robin order. The first client gets address 1, the next client gets address 2, and so on.

	Command or Action	Purpose
		• If this command is not configured, DHCP uses the server specified by the **ip helper address** command as the boot server.
Step 11	**netbios-name-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# netbios-name-server 172.16.1.103 172.16.2.103`	(Optional) Specifies the NetBIOS Windows Internet Naming Service (WINS) server that is available to a Microsoft DHCP client. • One address is required; however, you can specify up to eight addresses in one command line. • Servers should be listed in order of preference.
Step 12	**netbios-node-type** *type* **Example:** `Device(dhcp-config)# netbios-node-type h-node`	(Optional) Specifies the NetBIOS node type for a Microsoft DHCP client.
Step 13	**default-router** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# default-router 172.16.1.100 172.16.1.101`	(Optional) Specifies the IP address of the default device for a DHCP client. • The IP address should be on the same subnet as the client. • One IP address is required; however, you can specify a up to eight IP addresses in one command line. These default devices are listed in order of preference; that is, *address* is the most preferred device, *address2* is the next most preferred device, and so on. • When a DHCP client requests for an IP address, the device—acting as a DHCP server—accesses the default device list to select another device that the DHCP client uses as the first hop for forwarding messages. After a DHCP client has booted, the client begins sending packets to its default device.
Step 14	**option** *code* [**instance** *number*] {**ascii** *string* | **hex** *string* | *ip-address*} **Example:** `Device(dhcp-config)# option 19 hex 01`	(Optional) Configures DHCP server options.
Step 15	**lease** {*days* [*hours*] [*minutes*] | **infinite**} **Example:** `Device(dhcp-config)# lease 30`	(Optional) Specifies the duration of the lease. • The default is a one-day lease. • The **infinite** keyword specifies that the duration of the lease is unlimited.

	Command or Action	Purpose
Step 16	**network** *network-number* [*mask* | */prefix-length*] [**secondary**] **Example:** `Device(dhcp-config)# network 10.10.0.0 255.255.0.0 secondary`	(Optional) Specifies the network number and mask of a secondary DHCP server address pool. • Any number of secondary subnets can be added to a DHCP server address pool. • During execution of this command, the configuration mode changes to DHCP pool secondary subnet configuration mode, which is identified by (config-dhcp-subnet-secondary)# prompt. In this mode, the administrator can configure a default device list that is specific to the subnet. • See Troubleshooting Tips section if you are using secondary IP addresses under a loopback interface with DHCP secondary subnets.
Step 17	**override default-router** *address* [*address2 ... address8*] **Example:** `Device(config-dhcp-subnet-secondary)# override default-router 10.10.0.100 10.10.0.101`	(Optional) Specifies the default device list that is used when an IP address is assigned to a DHCP client from a particular secondary subnet. • If the subnet-specific override value is configured, this override value is used when assigning an IP address from the subnet; the network-wide default device list is used only to set the gateway device for the primary subnet. • If this subnet-specific override value is not configured, the network-wide default device list is used when assigning an IP address from the subnet. • See Example: Configuring a DHCP Address Pool with Multiple Disjoint Subnets section for a sample configuration.
Step 18	**override utilization high** *percentage-number* **Example:** `Device(config-dhcp-subnet-secondary)# override utilization high 60`	(Optional) Sets the high utilization mark of the subnet size. • This command overrides the global default setting specified by the **utilization mark high** command.
Step 19	**override utilization low** *percentage-number* **Example:** `Device(config-dhcp-subnet-secondary)# override utilization low 40`	(Optional) Sets the low utilization mark of the subnet size. • This command overrides the global default setting specified by the **utilization mark low** command.
Step 20	**end** **Example:** `Device(config-dhcp-subnet-secondary)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

If you are using secondary IP addresses under a single loopback interface and using secondary subnets under a DHCP pool, use one DHCP pool to configure networks for all the secondary subnets instead of using one pool per secondary subnet. The **network** *network-number* [*mask* | */prefix-length*] [**secondary**] command must be configured under a single DHCP address pool rather than multiple DHCP address pools.

The following is the correct configuration:

```
!
ip dhcp pool dhcp_1
 network 172.16.1.0 255.255.255.0
 network 172.16.2.0 255.255.255.0 secondary
 network 172.16.3.0 255.255.255.0 secondary
 network 172.16.4.0 255.255.255.0 secondary
!
interface Loopback111
 ip address 172.16.1.1 255.255.255.255 secondary
 ip address 172.16.2.1 255.255.255.255 secondary
 ip address 172.16.3.1 255.255.255.255 secondary
 ip address 172.16.4.1 255.255.255.255 secondary
```

The following is the incorrect configuration:

```
!
ip dhcp pool dhcp_1
 network 172.16.1.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_2
 network 172.16.2.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_3
 network 172.16.3.0 255.255.255.0
 lease 1 20 30
 accounting default
!
ip dhcp pool dhcp_4
 network 172.16.4.0 255.255.255.0
 lease 1 20 30
 accounting default
!
interface Loopback111
 ip address 172.16.1.1 255.255.255.255 secondary
 ip address 172.16.2.1 255.255.255.255 secondary
 ip address 172.16.3.1 255.255.255.255 secondary
 ip address 172.16.4.1 255.255.255.255 secondary
```

Verifying the DHCP Address Pool Configuration

The following configuration commands are optional. You can enter the **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **show ip dhcp pool** [*name*]
3. **show ip dhcp binding** [*address*]

4. **show ip dhcp conflict** [*address*]
5. **show ip dhcp database** [*url*]
6. **show ip dhcp server statistics** [*type-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip dhcp pool** [*name*] **Example:** `Device# show ip dhcp pool`	(Optional) Displays information about DHCP address pools.
Step 3	**show ip dhcp binding** [*address*] **Example:** `Device# show ip dhcp binding`	(Optional) Displays a list of all bindings created on a specific DHCP server. • Use the **show ip dhcp binding** command to display the IP addresses that have already been assigned. Verify that the address pool is not exhausted. If necessary, recreate the pool to create a larger pool of addresses. • Use the **show ip dhcp binding** command to display the lease expiration date and time of the IP address of the host.
Step 4	**show ip dhcp conflict** [*address*] **Example:** `Device# show ip dhcp conflict`	(Optional) Displays a list of all IP address conflicts.
Step 5	**show ip dhcp database** [*url*] **Example:** `Device# show ip dhcp database`	(Optional) Displays recent activity on the DHCP database.
Step 6	**show ip dhcp server statistics** [*type-number*] **Example:** `Device# show ip dhcp server statistics`	(Optional) Displays count information about server statistics and messages sent and received.

Configuring Manual Bindings

An address binding is a mapping between the IP address and MAC address of a client. The IP address of a client can be assigned manually by an administrator or assigned automatically from a pool by a DHCP server.

Manual bindings are IP addresses that are manually mapped to MAC addresses of hosts that are found in the DHCP database. Manual bindings are stored in the NVRAM of the DHCP server. Manual bindings are just special address pools. There is no limit to the number of manual bindings, but you can configure only one manual binding per host pool.

Automatic bindings are IP addresses that have been automatically mapped to MAC addresses of hosts that are found in the DHCP database. Because the bindings are stored in the volatile memory of the DHCP server, binding information is lost in the event of power failures or on device reloads. To prevent the loss of automatic binding information, a copy of the automatic binding information is stored on a remote host called the DHCP database agent. The bindings are periodically written to the database agent. When the device reloads, the bindings are read from the database agent to the DHCP database in the DHCP server.

Note We strongly recommend that you use database agents. However, Cisco DHCP server can function even without database agents.

Some DHCP clients send a client identifier (DHCP option 61) in the DHCP packet. To configure manual bindings for such clients, you must enter the **client-identifier** command with the hexadecimal values that identify the DHCP client. To configure manual bindings for clients that do not send a client identifier option, you must enter the **hardware-address** DHCP pool configuration command with the hexadecimal hardware address of the client.

Depending on your release, the DHCP server sends infinite lease time to the clients for which manual bindings are configured.

Depending on your release, the DHCP server sends lease time that is configured using the **lease** command to clients for which manual bindings are configured.

Note You cannot configure manual bindings within the same pool that is configured with the **network** command in DHCP pool configuration mode. See the Configuring DHCP Address Pools section for information about DHCP address pools and the **network** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **host** *address* [*mask* | */prefix-length*]
5. **client-identifier** *unique-identifier*
6. **hardware-address** *hardware-address* [*protocol-type* | *hardware-number*]
7. **client-name** *name*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Device(config)# ip dhcp pool pool1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.
Step 4	**host** *address* [*mask* | */prefix-length*] **Example:** `Device(dhcp-config)# host 172.16.0.1`	Specifies the IP address and subnet mask of the client. • There is no limit to the number of manual bindings you can configure. However, you can configure only one manual binding per host pool.

Step 5	**client-identifier** *unique-identifier*
	Example:
	`Device(dhcp-config)# client-identifier 01b7.0813.8811.66`
		Specifies the unique identifier for DHCP clients.
		• This command is used for DHCP requests.
		• DHCP clients require client identifiers. You can specify the unique identifier for the client in either of the following ways:
		  • A 7-byte dotted hexadecimal notation. For example, 01b7.0813.8811.66, where 01 represents the Ethernet media type and the remaining bytes represent the MAC address of the DHCP client.
		  • A 27-byte dotted hexadecimal notation. For example, 7665e646f722d30302342e3937623032e333734312d466130231. The equivalent ASCII string for this hexadecimal value is vendor-0024.97b0.3741-fa0/1, where vendor represents the vendor, 0024.97b0.3741 represents the MAC address of the source interface, and fa0/1 represents the source interface of the DHCP client.
		• See the Troubleshooting section for information about how to determine the client identifier of the DHCP client.

	Command or Action	Purpose
		Note The identifier specified here is considered for a DHCP client that sends a client identifier in the packet.
Step 6	**hardware-address** *hardware-address* [*protocol-type* | *hardware-number*] **Example:** `Device(dhcp-config)# hardware-address b708.1388.f166 ethernet`	Specifies a hardware address for the client. • This command is used for BOOTP requests. **Note** The hardware address specified here is considered for a DHCP client that does not send a client identifier in the packet.
Step 7	**client-name** *name* **Example:** `Device(dhcp-config)# client-name client1`	(Optional) Specifies the name of the client using any standard ASCII character. • The client name should not include the domain name. For example, the name client1 should not be specified as client1.cisco.com.
Step 8	**end** **Example:** `Device(dhcp-config)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

Use the following command to debug any errors that you may encounter when you configure DHCP to automatically generate a unique ID:

- **debug ip dhcp server packets**

Configuring DHCP Static Mapping

The DHCP Static Mapping feature enables the assignment of static IP addresses (without creating numerous host pools with manual bindings) by using a customer-created text file that the DHCP server reads. The benefit of this feature is that it eliminates the need for a long configuration file and reduces the space required in NVRAM to maintain address pools.

A DHCP database contains the mappings between a client IP address and the hardware address, which is referred to as a binding. There are two types of bindings: manual bindings that map a single hardware address to a single IP address, and automatic bindings that dynamically map a hardware address to an IP address from a pool of IP addresses. Manual (also known as static) bindings can be configured individually directly on the device or by using the DHCP Static Mapping feature. These static bindings can be read from a separate static mapping text file. The static mapping text files are read when a device reloads or the DHCP service restarts. These files are read-only.

The read static bindings are treated just like the manual bindings, in that they are:

- Retained across DHCPRELEASEs from the clients.

- Not timed out.
- Deleted only upon deletion of the pool.
- Provided appropriate exclusions for the contained addresses, which are created at the time of the read.

Just like automatic bindings, manual (or static) bindings from the static mapping text file are also displayed by using the **show ip dhcp binding** command.

Perform this task to create the static mapping text file. You will input your addresses in the text file, which is stored in the DHCP database for the DHCP server to read. There is no limit to the number of addresses that can be stored in the file. The file format has the following elements:

- Database version number
- End-of-file designator
- Hardware type
- Hardware address
- IP address
- Lease expiration
- Time the file was created

See the following table for more details about the format of the text file.

The following is a sample static mapping text file:

```
*time* Jan 21 2005 03:52 PM
*version* 2
!IP address    Type    Hardware address     Lease expiration
10.0.0.4 /24   1       0090.bff6.081e       Infinite
10.0.0.5 /28   id      00b7.0813.88f1.66    Infinite
10.0.0.2 /21   1       0090.bff6.081d       Infinite
*end*
```

Table 2: Static Mapping Text File Field Descriptions

Field	Description
time	Specifies the time the file was created. This field allows DHCP to differentiate between the new and old database versions when multiple agents are configured. The valid format of the time is mm dd yyyy hh:mm AM/PM.
version 2	Specifies the database version number.
IP address	Specifies the static IP address. If the subnet mask is not specified, a mask is automatically assigned depending on the IP address. The IP address and the mask is separated by a space.
Type	Specifies the hardware type. For example, type “1” indicates Ethernet. The type “id” indicates that the field is a DHCP client identifier. Legal values can be found online at http://www.iana.org/assignments/arp-parameters/arp-parameters.xhtml in the “Number Hardware Type” list.

Field	Description
Hardware address	Specifies the hardware address. When the type is numeric, the type refers to the hardware media. Legal values can be found online at http://www.iana.org/assignments/arp-parameters/arp-parameters.xhtml in the "Number Hardware Type" list. When the type is "id," the type refers to a match on the client identifier. For more information about the client identifier, see RFC 2132, *DHCP Options and BOOTP Vendor Extensions*, section 9.14, located at https://www.ietf.org/rfc/rfc2132.txt, or the **client-identifier** command. If you are unsure about the client identifier to match with the hardware type, use the **debug dhcp detail** command to display the client identifier being sent to the DHCP server from the client.
Lease expiration	Specifies the expiration of the lease. "Infinite" specifies that the duration of the lease is unlimited.
end	End of file. DHCP uses the *end* designator to detect file truncation.

Configuring the DHCP Server to Read a Static Mapping Text File

Before you begin

The administrator must create the static mapping text file in the correct format and configure the address pools before performing this task.

Before editing the file, you must disable the DHCP server using the **no service dhcp** command.

Note The static bindings must not be deleted when a DHCPRELEASE is received or must not be timed out by the DHCP timer. The static bindings should be created by using the **ip dhcp pool** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **origin file** *url*
5. **end**
6. **show ip dhcp binding** [*address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Device(config)# ip dhcp pool pool1`	Assigns a name to a DHCP pool and enters DHCP configuration mode. **Note** If you have already configured the IP DHCP pool name using the **ip dhcp pool** command and the static file URL using the **origin file** command, you must perform a fresh read using the **no service dhcp** command and the **service dhcp** command.
Step 4	**origin file** *url* **Example:** `Device(dhcp-config)# origin file tftp://10.1.0.1/static-bindings`	Specifies the URL that the DHCP server can access to locate the text file.
Step 5	**end** **Example:** `Device(dhcp-config)# end`	Returns to privileged EXEC mode.
Step 6	**show ip dhcp binding** [*address*] **Example:** `Device# show ip dhcp binding`	(Optional) Displays a list of all bindings created on a specific DHCP server.

Examples

The following sample output from the **show ip dhcp binding** command displays address bindings that are configured:

```
Device# show ip dhcp binding

00:05:14:%SYS-5-CONFIG_I: Configured from console by console
Bindings from all pools not associated with VRF:
IP address  Client-ID/            Ls expir   Type    Hw address           User name
10.9.9.4/8  0063.7363.2d30.3036.  Infinite   Static  302e.3762.2e39.3634. 632d.4574.8892.
10.9.9.1/24 0063.6973.636f.2d30.  Infinite   Static  3036.302e.3437.3165. 2e64.6462.342d.
```

The following sample output displays each entry in the static mapping text file:

```
*time* Jan 21 2005 22:52 PM
```

```
!IP address     Type        Hardware address                              Lease expiration
10.19.9.1 /24    id          0063.6973.636f.2d30.3036.302e.3437
10.9.9.4         id          0063.7363.2d30.3036.302e.3762.2e39.3634.632d  Infinite
*end*
```

The following sample debug output shows the reading of the static mapping text file from the TFTP server:

```
Device# debug ip dhcp server

Loading abc/static_pool from 10.19.192.33 (via Ethernet0):
[OK - 333 bytes]
*May 26 23:14:21.259: DHCPD: contacting agent tftp://10.19.192.33/abc/static_pool (attempt
 0)
*May 26 23:14:21.467: DHCPD: agent tftp://10.19.192.33/abc/static_pool is responding.
*May 26 23:14:21.467: DHCPD: IFS is ready.
*May 26 23:14:21.467: DHCPD: reading bindings from tftp://10.19.192.33/abc/static_pool.
*May 26 23:14:21.707: DHCPD: read 333 / 1024 bytes.
*May 26 23:14:21.707: DHCPD: parsing text line
*time* Apr 22 2002 11:31 AM
*May 26 23:14:21.707: DHCPD: parsing text line ""
*May 26 23:14:21.707: DHCPD: parsing text line
!IP address Type Hardware address Lease expiration
*May 26 23:14:21.707: DHCPD: parsing text line
"10.9.9.1 /24 id 0063.6973.636f.2d30.3036.302e.3437"
*May 26 23:14:21.707: DHCPD: creating binding for 10.9.9.1
*May 26 23:14:21.707: DHCPD: Adding binding to radix tree (10.9.9.1)
*May 26 23:14:21.707: DHCPD: Adding binding to hash tree
*May 26 23:14:21.707: DHCPD: parsing text line
"10.9.9.4 id 0063.7363.2d30.3036.302e.3762.2e39.3634.632d"
*May 26 23:14:21.711: DHCPD: creating binding for 10.9.9.4
*May 26 23:14:21.711: DHCPD: Adding binding to radix tree (10.9.9.4)
*May 26 23:14:21.711: DHCPD: Adding binding to hash tree
*May 26 23:14:21.711: DHCPD: parsing text line "Infinite"
*May 26 23:14:21.711: DHCPD: parsing text line ""
*May 26 23:14:21.711: DHCPD: parsing text line
!IP address Interface-index Lease expiration VRF
*May 26 23:14:21.711: DHCPD: parsing text line "*end*"
*May 26 23:14:21.711: DHCPD: read static bindings from tftp://10.19.192.33/abcemp/static_pool.
```

Customizing DHCP Server Operation

By default, the DHCP server pings a pool address twice before assigning a particular address to a requesting client. If the ping is unanswered, the DHCP server assumes (with a high probability) that the address is not in use and assigns the address to the requesting client.

By default, the DHCP server waits for 2 seconds before timing out a ping packet.

You can configure the DHCP server to ignore and not reply to any BOOTP requests that the server receives. This functionality is beneficial when there is a mix of BOOTP and DHCP clients in a network segment and there is a BOOTP server and a Cisco DHCP server servicing the network segment. The BOOTP server is configured with static bindings for the BOOTP clients and the BOOTP clients must obtain their addresses from the BOOTP server. However, DHCP servers can also respond to BOOTP requests and the DHCP server may offer an address that causes the BOOTP clients to boot with the address from the DHCP server, instead of the address from the BOOTP server. Configuring the DHCP server to ignore BOOTP requests ensures that the BOOTP clients will receive address information from the BOOTP server and will not accept an address from a DHCP server.

Cisco software can forward these ignored BOOTP request packets to another DHCP server if the **ip helper-address** command is configured on the incoming interface.

Note It is not recommended to use DHCP ping checks on Cisco Catalyst switches implemented in switch stack or VSS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp ping packets** *number*
4. **ip dhcp ping timeout** *milliseconds*
5. **ip dhcp bootp ignore**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp ping packets** *number* **Example:** `Device(config)# ip dhcp ping packets 5`	(Optional) Specifies the number of ping packets the DHCP server sends to a pool address before assigning the address to a requesting client. • The default is two packets. Setting the *number* argument to a value of 0 disables the DHCP server ping operation.
Step 4	**ip dhcp ping timeout** *milliseconds* **Example:** `Device(config)# ip dhcp ping timeout 850`	(Optional) Specifies the duration the DHCP server waits for a ping reply from an address pool.
Step 5	**ip dhcp bootp ignore** **Example:** `Device(config)# ip dhcp bootp ignore`	(Optional) Allows the DHCP server to selectively ignore and not reply to received BOOTP requests. • The **ip dhcp bootp ignore** command applies to all DHCP pools configured on the device. BOOTP requests cannot be selectively ignored on a per-DHCP pool basis.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring a Remote Device to Import DHCP Server Options from a Central DHCP Server

The Cisco DHCP server can dynamically configure options such as the Domain Name System (DNS) and Windows Internet Name Service (WINS) addresses to respond to DHCP requests from local clients behind the customer premises equipment (CPE). Earlier, network administrators configured the Cisco DHCP server on each device manually. Now, the Cisco DHCP server is enhanced to allow configuration information to be updated automatically. Network administrators can configure one or more centralized DHCP servers to update specific DHCP options within the DHCP pools. The remote servers can request or "import" these option parameters from centralized servers.

This section contains the following tasks:

Configuring the Central DHCP Server to Update DHCP Options

Perform the following task to configure the Central DHCP Server to update DHCP options:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **network** *network-number* [*mask* | */prefix-length*]
5. **dns-server** *address* [*address2 ... address8*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:**	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.

	Command or Action	Purpose
	`Device(config)# ip dhcp pool 1`	
Step 4	**network** *network-number* [*mask* | */prefix-length*] **Example:** `Device(dhcp-config)# network 172.16.0.0 /16`	Specifies the subnet number and mask of the DHCP address pool.
Step 5	**dns-server** *address* [*address2 ... address8*] **Example:** `Device(dhcp-config)# dns server 172.16.1.103 172.16.2.103`	(Optional) Specifies the IP address of a DNS server that is available to a DHCP client. • One IP address is required; however, you can specify up to eight IP addresses in one command line. • Servers should be listed in the order of preference.
Step 6	**end** **Example:** `Device(dhcp-config)# end`	Returns to privileged EXEC mode.

Configuring the Remote Device to Import DHCP Options

Perform the following task to configure the remote device to import DHCP options:

Note When two servers provide DHCP addresses to a single device configured with **ip address dhcp** on two different interfaces, the imported information is merged and, for those options that take a single value, the last known option value will be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *pool-name*
4. **network** *network-number* [*mask* | */prefix-length*]
5. **import** {**all** | **interface** *interface_name*
6. **exit**
7. **interface** *type number*
8. **ip address dhcp**
9. **end**
10. **show ip dhcp import**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *pool-name* **Example:** `Device(config)# ip dhcp pool pool1`	Creates a name for the DHCP server address pool and enters DHCP pool configuration mode.
Step 4	**network** *network-number* [*mask* | */prefix-length*] **Example:** `Device(dhcp-config)# network 172.30.0.0 /16`	Specifies the subnet network number and mask of the DHCP address pool.

Step 5	**import** {**all** | **interface** *interface_name*
	Example:
	`Device(dhcp-config)# import all`
	`Device(dhcp-config) # import interface Ethernet0/0`
		Imports DHCP option parameters into the DHCP server database.

Step 6	**exit** **Example:** `Device(dhcp-config)# exit`	Exits DHCP pool configuration mode and enters global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Configures an interface and enters interface configuration mode.
Step 8	**ip address dhcp** **Example:** `Device(config-if)# ip address dhcp`	Specifies that the interface acquires an IP address through DHCP.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 10	**show ip dhcp import** **Example:**	Displays the options that are imported from the central DHCP server.

	Command or Action	Purpose
	`Device# show ip dhcp import`	

Configuring DHCP Address Allocation Using Option 82

Restrictions for DHCP Address Allocation Using Option 82

If the relay agent inserts option 82 but does not set the giaddr field in the DHCP packet, the DHCP server interface must be configured as a trusted interface by using the **ip dhcp relay information trusted** global configuration command. This configuration prevents the server from dropping the DHCP message.

Enabling Option 82 for DHCP Address Allocation

By default, the Cisco DHCP server uses information provided by option 82 to allocate IP addresses. If the DHCP address allocation is disabled, perform the task described in this section to reenable this capability.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp use class**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp use class** **Example:** `Device(config)# ip dhcp use class`	Controls DHCP classes that are used for address allocation. • This functionality is enabled by default. • Use the **no** form of this command to disable this functionality without deleting the DHCP class configuration.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

If DHCP classes are configured in the pool, but the DHCP server does not use the classes, verify if the **no ip dhcp use class** command was configured.

Defining the DHCP Class and Relay Agent Information Patterns

Before you begin

You must know the hexadecimal value of each byte location in option 82 to configure the **relay-information hex** command. The option 82 format may vary from product to product. Contact the relay agent vendor for this information.

Perform this task to define the DHCP class and relay agent information patterns:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp class** *class-name*
4. **relay agent information**
5. **relay-information hex** *pattern* [*] [**bitmask** *mask*]
6. Repeat Steps 3 through 5 for each DHCP class you need to configure.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp class** *class-name* **Example:** `Device(config)# ip dhcp class CLASS1`	Defines a DHCP class and enters DHCP class configuration mode.
Step 4	**relay agent information** **Example:** `Device(dhcp-class)# relay agent information`	Enters relay agent information option configuration mode. • If you omit this step, the DHCP class matches any relay agent information option, whether the relay agent information option value is available or not.

	Command or Action	Purpose
Step 5	**relay-information hex** *pattern* [*] [**bitmask** *mask*] **Example:** `Device(dhcp-class-relayinfo)# relay-information hex 01030a0b0c02050000000123`	(Optional) Specifies a hexadecimal value for full relay information option. • The *pattern* argument creates a pattern that is used to match the DHCP class. • If you omit this step, no pattern is configured and it is considered a match to any relay agent information option value, but the relay information option must be available in the DHCP packet. • You can configure multiple **relay-information hex** commands in a DHCP class.
Step 6	Repeat Steps 3 through 5 for each DHCP class you need to configure.	
Step 7	**end** **Example:** `Device(dhcp-class-relayinfo)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

If DHCP classes are configured in the pool, but the DHCP server does not use the classes, verify if the **no ip dhcp use class** command was configured.

Defining the DHCP Address Pool

Perform this task to define the DHCP address pool:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp pool** *name*
4. **network** *network-number* [*mask* | */prefix-length*]
5. **class** *class-name*
6. **address range** *start-ip end-ip*
7. Repeat Steps 5 and 6 for each DHCP class you need to associate with the DHCP pool.
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp pool** *name* **Example:** `Device# ip dhcp pool ABC`	Configures a DHCP address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode. • Multiple pools can be configured with the same class, eliminating the need to configure the same pattern in multiple pools.
Step 4	**network** *network-number* [*mask* | */prefix-length*] **Example:** `Device(dhcp-config)# network 10.0.20.0`	Configures the subnet and mask for a DHCP address pool on a Cisco IOS DHCP server.
Step 5	**class** *class-name* **Example:** `Device(dhcp-config)# class CLASS1`	Associates a class with a pool and enters DHCP pool class configuration mode. • This command also creates a DHCP class if the DHCP class is not yet defined.
Step 6	**address range** *start-ip end-ip* **Example:** `Device(dhcp-pool-class)# address range 10.0.20.1 10.0.20.100`	(Optional) Sets an address range for the DHCP class in a DHCP server address pool. • If this command is not configured for a class, the default value is the entire subnet of the pool. Each class in the DHCP pool is examined for a match in the order configured.
Step 7	Repeat Steps 5 and 6 for each DHCP class you need to associate with the DHCP pool.	
Step 8	**end** **Example:** `Device(dhcp-pool-class)# end`	Returns to privileged EXEC mode.

Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP

Perform this task to configure a static route to use a DHCP default gateway as the next-hop router.

This task enables static routes to be assigned using a DHCP default gateway as the next-hop router. This behavior was not possible before the introduction of this feature because the gateway IP address is not known

until after the DHCP address assignment. A static route could not be configured with the command-line interface (CLI) that used that DHCP-supplied address.

The static routes are installed in the routing table when the default gateway is assigned by the DHCP server. The routes remain in the routing table until the DHCP lease expires at which time the routes are removed.

When a DHCP client releases an address, the corresponding static route (the route configured with the **ip route**command) is automatically removed from the routing table. If the DHCP router option (option 3 of the DHCP packet) changes during the client renewal, the DHCP default gateway changes to the new IP address supplied in the renewal.

This feature is particularly useful for VPN deployments such as Dynamic Multipoint VPNs (DMVPNs). This feature is useful when a non-physical interface like a multipoint generic routing encapsulation (mGRE) tunnel is configured on the router and certain traffic needs to be excluded from going to the tunnel interface.

Before you begin

Verify all DHCP client and server configuration steps. Ensure that the DHCP client and server are properly defined to supply a DHCP router option 3.

Note

- If the DHCP client is not able to obtain an IP address or default router IP address, the static route is not installed in the routing table.
- If the lease has expired and the DHCP client cannot renew the address, the DHCP IP address assigned to the client is released and any associated static routes are removed from the routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **dhcp** [*distance*]
4. **end**
5. **show ip route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **dhcp** [*distance*] **Example:**	Assigns a static route for the default next-hop router when the DHCP server is accessed for an IP address.

	Command or Action	Purpose

	`Device(config)# ip route 209.165.200.225 255.255.255.255 GigabitEthernet 0/0/0 dhcp`
	Example:
	`Device(config)# ip route 209.165.200.226 255.255.255.255 GigabitEthernet 0/0/1 dhcp 20`
		• If more than one interface on a router is configured to obtain an IP address from a DHCP server, use the **ip route** *prefix mask interface-type interface-number* **dhcp** command for each interface. If the interface is not specified, the route is added to the routing table as soon as any of the interfaces obtain an IP address and default router.

Step 4	**end** **Example:** `Device(config)# end`	Returns to global configuration mode.
Step 5	**show ip route** **Example:** `Device# show ip route`	(Optional) Displays the current state of the routing table. • Use this command to display assigned static routes once the DHCP client obtains an address and a default router address from the DHCP server.

Clearing DHCP Server Variables

Perform this task to clear DHCP server variables:

SUMMARY STEPS

1. **enable**
2. **clear ip dhcp binding** {*address* | *****}
3. **clear ip dhcp conflict** {*address* | *****}
4. **clear ip dhcp server statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip dhcp binding** {*address*	*****} **Example:** `Device# clear ip dhcp binding *`	Deletes an automatic address binding from the DHCP database. • Specifying the *address* argument clears the automatic binding for a specific (client) IP address, whereas specifying an asterisk (*****) clears all automatic bindings.
Step 3	**clear ip dhcp conflict** {*address*	*****} **Example:**	Clears an address conflict from the DHCP database.

	Command or Action	Purpose
	`Device# clear ip dhcp conflict 172.16.1.103`	• Specifying the *address* argument clears the conflict for a specific IP address, whereas specifying an asterisk (*****) clears conflicts for all addresses.
Step 4	**clear ip dhcp server statistics** **Example:** `Device# clear ip dhcp server statistics`	Resets all DHCP server counters to 0.

Configuration Examples for the Cisco IOS XE DHCP Server

Example: Configuring the DHCP Database Agent

The following example shows how to store bindings on host 172.16.4.253. The file transfer protocol is FTP. The server waits for 2 minutes (120 seconds) before performing database changes.

```
ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
```

Example: Excluding IP Addresses

In the following example, server A and server B service the subnet 10.0.20.0/24. If the subnet is split equally between the two servers, server A is configured to allocate IP addresses 10.0.20.1 to 10.0.20.125 and server B is configured to allocate IP addresses 10.0.20.126 to 10.0.20.254.

Server A

```
ip dhcp excluded-address 10.0.20.126 10.0.20.255
!
ip dhcp pool A
 network 10.0.20.0 255.255.255.0
```

Server B

```
ip dhcp excluded-address 10.0.20.0 10.0.20.125
!
ip dhcp pool B
 network 10.0.20.0 255.255.255.0
```

Example: Configuring DHCP Address Pools

In the following example, three DHCP address pools are created: one in network 172.16.0.0, one in subnetwork 172.16.1.0, and one in subnetwork 172.16.2.0. Attributes from network 172.16.0.0—such as the domain name, Domain Name System (DNS) server, (Network Basic Input/Output System) NetBIOS name server, and NetBIOS node type—are inherited in subnetworks 172.16.1.0 and 172.16.2.0. In each pool, clients are granted 30-day leases and all addresses in each subnetwork, except the excluded addresses, are available to the DHCP

server for assigning to clients. The table below lists the IP addresses for the devices in three DHCP address pools.

Table 3: DHCP Address Pool Configuration

Pool 0 (Network 172.16.0.0)	**Pool 1 (Subnetwork 172.16.1.0)**	**Pool 2 (Subnetwork 172.16.2.0)**			
Device	IP Address	Device	IP Address	Device	IP Address
Default devices	—	Default devices	172.16.1.100 172.16.1.101	Default devices	172.16.2.100 172.16.2.101
DNS server	172.16.1.102 172.16.2.102	—	—	—	—
NetBIOS name server	172.16.1.103 172.16.2.103	—	—	—	—
NetBIOS node type	h-node	—	—	—	—

```
ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
ip dhcp excluded-address 172.16.1.100 172.16.1.103
ip dhcp excluded-address 172.16.2.100 172.16.2.103
!
ip dhcp pool 0
 network 172.16.0.0 /16
 domain-name cisco.com
 dns-server 172.16.1.102 172.16.2.102
 netbios-name-server 172.16.1.103 172.16.2.103
 netbios-node-type h-node
!
ip dhcp pool 1
 network 172.16.1.0 /24
 default-router 172.16.1.100 172.16.1.101
 lease 30
!
ip dhcp pool 2
 network 172.16.2.0 /24
 default-router 172.16.2.100 172.16.2.101
 lease 30
```

The following example shows how to configure DHCP pool to support RegEx feature:

```
!
ip dhcp pool test
network 192.168.10.0 255.255.255.0
default-router 192.168.10.1
 dns-server 40.0.0.100
 class cisco_devices
  address range 192.168.10.2 192.168.10.100
!
class smart_phones
  address range 192.168.10.101 192.168.10.220
!
!
ip dhcp class cisco_devices
 option 60 cisco_string ----------------------<this is option 60 VCI string, exact match>
```

```
!
ip dhcp class smart_phones
 option 60 smartphone* ----------------------<option 60 VCI string, regex match>
!
```

The following example shows how to configure DHCP server class:

```
Router#
Router# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip dhcp class HATHWAY_STB
Router(config-dhcp-class)#?
DHCP class configuration commands:
  exit        Exit from DHCP class configuration mode
  no         Negate a command or set its defaults
 option   Raw DHCP options
  relay    Enter relay agent information option configuration submode
  remark  Specify a remark for this class

Router(config-dhcp-class)#option ?
  <0-254>  DHCP option code

Router(config-dhcp-class)#option 60 ?
  hex  Specify hex value of the option
  WORD  Specify a regular expression string

Router(config-dhcp-class)#option 60 stb*  ?
<cr>
```

The following example shows how to Import options learnt on specific interface to LAN side DHCP pool:

```
!
ip dhcp pool LAN_Pool
import interface Ethernet0/0
!

Router#
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# ip dhcp pool pc_pool
Router(dhcp-config)# import ?
  all              all DHCP options
  interface   Select an interface to import options
Router(dhcp-config)# import interface Ethernet0/1
```

Example: Configuring a DHCP Address Pool with Multiple Disjoint Subnets

Multiple disjoint subnets in a DHCP pool can be used in any of the following network topologies:

- IP address pooling—The DHCP client and server reside on the same subnet.
- DHCP relay—The DHCP client and DHCP server communicate through a DHCP relay agent where the relay interface is configured with secondary IP addresses.
- Hierarchical DHCP—The DHCP server is configured as the DHCP subnet allocation server. The DHCP client and DHCP subnet allocation server communicate through an on-demand address pool (ODAP) router.

In the following example, one DHCP address pool named pool3 is created; the primary subnet is 172.16.0.0/16, one secondary subnet is 172.16.1.0/24, and the other secondary subnet is 172.16.2.0/24.

- When IP addresses in the primary subnet are exhausted, the DHCP server inspects the secondary subnets in the order in which the subnets were added to the pool.
- When the DHCP server allocates an IP address from the secondary subnet 172.16.1.0/24, the server uses the subnet-specific default device list that consists of IP addresses 172.16.1.100 and 172.16.1.101. However, when the DHCP server allocates an IP address from the subnet 172.16.2.0/24, the server uses the pool-wide list that consists of the four IP addresses from 172.16.0.100 to 172.16.0.103.
- Other attributes from the primary subnet 172.16.0.0/16—such as the domain name, DNS server, NetBIOS name server, and NetBIOS node type—are inherited in both the secondary subnets.
- DHCP clients are granted 30-day leases on IP addresses in the pool. All addresses in each subnet, except the excluded addresses, are available to the DHCP server for assigning to clients.

The table below lists the IP addresses for the devices in the DHCP address pool that consists of three disjoint subnets.

Table 4: DHCP Address Pool Configuration with Multiple Disjoint Subnets

Primary Subnet (172.16.0.0/16)	**First Secondary Subnet (172.16.1.0/24)**	**Second Secondary Subnet (172.16.2.0/24)**			
Device	IP Address	Device	IP Address	Device	IP Address
Default devices	172.16.0.100 172.16.0.101 172.16.0.102 172.16.0.103	Default devices	172.16.1.100 172.16.1.101	Default devices	172.16.0.100 172.16.0.101 172.16.0.102 172.16.0.103
DNS server	172.16.1.102 172.16.2.102	—	—	—	—
NetBIOS name server	172.16.1.103 172.16.2.103	—	—	—	—
NetBIOS node type	h-node	—	—	—	—

```
ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
ip dhcp excluded-address 172.16.0.100 172.16.1.103
ip dhcp excluded-address 172.16.1.100 172.16.1.101
!
ip dhcp pool pool3
 network 172.16.0.0 /16
 default-router 172.16.0.100 172.16.2.101 172.16.0.102 172.16.0.103
 domain-name cisco.com
 dns-server 172.16.1.102 172.16.2.102
 netbios-name-server 172.16.1.103 172.16.2.103
 netbios-node-type h-node
 lease 30
!
 network 172.16.1.0 /24 secondary
  override default-router 172.16.1.100 172.16.1.101
  end
```

```
!
 network 172.16.2.0 /24 secondary
```

Configuring Manual Bindings Example

The following example shows how to create a manual binding for a client named Mars.cisco.com. The MAC address of the client is 02c7.f800.0422 and the IP address of the client is 172.16.2.254.

```
ip dhcp pool Mars
 host 172.16.2.254
 hardware-address 02c7.f800.0422 ieee802
 client-name Mars
```

Because attributes are inherited, the previous configuration is equivalent to the following:

```
ip dhcp pool Mars
 host 172.16.2.254 mask 255.255.255.0
 hardware-address 02c7.f800.0422 ieee802
 client-name Mars
 default-router 172.16.2.100 172.16.2.101
 domain-name cisco.com
 dns-server 172.16.1.102 172.16.2.102
 netbios-name-server 172.16.1.103 172.16.2.103
 netbios-node-type h-node
```

Example: Configuring Static Mapping

The following example shows how to restart the DHCP server, configure the pool, and specify the URL where the static mapping text file is stored:

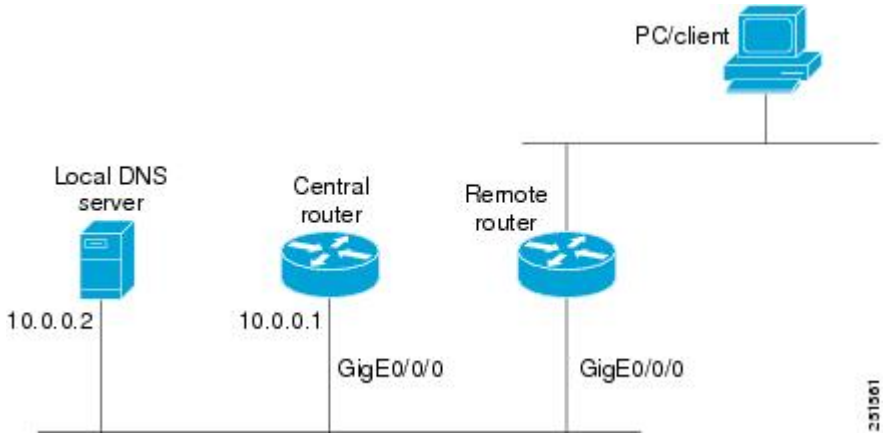
```
no service dhcp
service dhcp
ip dhcp pool abcpool
origin file tftp://10.1.0.1/staticfilename
```

Note The static mapping text file can be copied to flash memory on the device and served by the TFTP process of the device. In this case, the IP address in the original file line must be an address owned by the device and one additional line of configuration is required on the device:**tftp-server flash** *static-filename*.

Importing DHCP Options Example

The following example shows a remote and central server configured to support the importing of DHCP options. The central server is configured to automatically update DHCP options, such as DNS and WINs addresses, within the DHCP pools. In response to a DHCP request from a local client behind CPE equipment, the remote server can request or “import” these option parameters from the centralized server. See the figure below for a diagram of the network topology.

Figure 2: DHCP Example Network Topology



Central Router

```
!do not assign this range to DHCP clients
ip dhcp-excluded address 10.0.0.1 10.0.0.5
!
ip dhcp pool central
! Specifies network number and mask for DHCP clients
 network 10.0.0.0 255.255.255.0
! Specifies the domain name for the client
 domain-name central
! Specifies DNS server that will respond to DHCP clients when they need to correlate host
! name to ip address
 dns-server 10.0.0.2
!Specifies the NETBIOS WINS server
 netbios-name-server 10.0.0.2
!
interface GigabitEthernet0/0/0
 ip address 10.0.0.1 255.255.255.0
 duplex auto
 speed auto
```

Remote Router

```
ip dhcp pool client
! Imports DHCP option parameters into DHCP server database
 import all
 network 20.0.0.0 255.255.255.0
!
interface GigabitEthernet0/0/0
 ip address dhcp
 duplex auto
 speed auto
```

Configuring DHCP Address Allocation Using Option 82 Example

This example configures two DHCP classes. CLASS1 defines the group of DHCP clients whose address requests contain the relay agent information option with the specified hexadecimal values. CLASS2 defines the group of DHCP clients whose address requests contain the configured relay agent information suboptions.

CLASS3 has no pattern configured and is treated as a “match to any” class. This type of class is useful for specifying a “default” class.

In the following example, the subnet of pool ABC has been divided into three ranges without further subnetting of the 10.0.20.0/24 subnet. If there is a DHCP Discover message from the 10.0.20.0/24 subnet with option 82 matching that of class CLASS1, an available address in the range from 10.0.20.1 to 10.0.20.100 will be allocated. If there is no free address in CLASS1's address range, the DHCP Discover message will be matched against CLASS2, and so on.

Thus, each class in the DHCP pool will be examined for a match in the order configured by the user. In pool ABC, the order of matching is CLASS1, CLASS2, and finally CLASS3. In pool DEF, class CLASS2 does not have any address range configured. By default, the address range for a particular class is the pool's entire subnet(s). Therefore, clients matching CLASS2 may be allocated addresses from 11.0.20.1 to 11.0.20.254.

Multiple pools can be configured with the same class, eliminating the need to configure the same patterns in multiple pools. In the future, further classification method may be implemented. For example, there may be a need to specify that one or more pools should only be used to service a particular class of devices (for example, cable modems and IP phones).

```
! Defines the DHCP classes and relay information patterns
ip dhcp class CLASS1
 relay agent information
  relay-information hex 01030a0b0c02050000000123
  relay-information hex 01030a0b0c02*
  relay-information hex 01030a0b0c02050000000000 bitmask 000000000000000000000FF
ip dhcp class CLASS2
 relay agent information
  relay-information hex 01040102030402020102
  relay-information hex 01040101030402020102
ip dhcp class CLASS3
 relay agent information
! Associates the DHCP pool with DHCP classes
ip dhcp pool ABC
 network 10.0.20.0 255.255.255.0
 class CLASS1
  address range 10.0.20.1 10.0.20.100
class CLASS2
  address range 10.0.20.101 10.0.20.200
 class CLASS3
  address range 10.0.20.201 10.0.20.254
ip dhcp pool DEF
 network 11.0.20.0 255.255.255.0
 class CLASS1
  address range 11.0.20.1 11.0.20.64
 class CLASS2
```

Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP Example

The following example shows how to configure two GigabitEthernet interfaces to obtain the next-hop router IP address from the DHCP server:

```
ip route 10.10.10.0 255.255.255.0 dhcp 200
ip route 10.10.20.1 255.255.255.255 gigaether 1 dhcp
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Command List, All Releases
DHCP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
DHCP conceptual information	"DHCP Overview" module
DHCP relay agent configuration	"Configuring the Cisco IOS XE DHCP Relay Agent" module
DHCP client configuration	"Configuring the Cisco IOS XE DHCP Client" module
DHCP On-Demand Address Pool Manager	"Configuring the DHCP On-Demand Address Pool Manager" module

Standards and RFCs

Standard/RFC	Title
RFC 951	*Bootstrap Protocol (BOOTP)*
RFC 1542	*Clarifications and Extensions for the Bootstrap Protocol*
RFC 2131	*Dynamic Host Configuration Protocol*
RFC 2132	*DHCP Options and BOOTP Vendor Extensions*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	https://www.cisco.com/c/en/us/support/index.html

Feature Information for the Cisco IOS XE DHCP Server

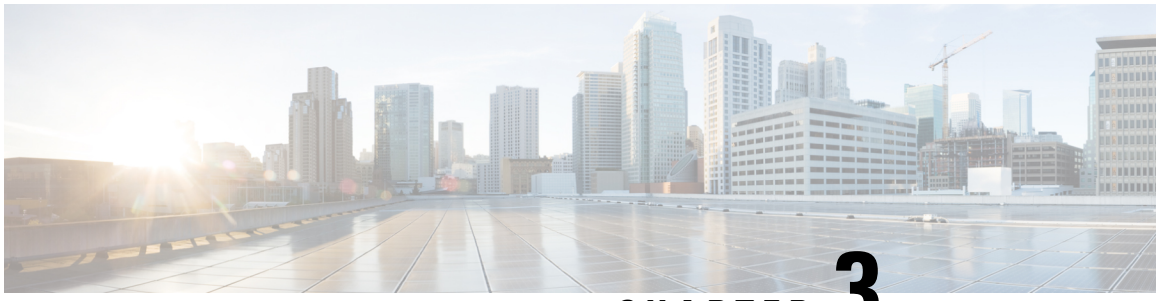
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for the Cisco IOS XE DHCP Server

Feature Name	Releases	Feature Configuration Information
DHCP Server	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.9S	Cisco routers running Cisco IOS XE software include Dynamic Host Configuration Protocol (DHCP) server and relay agent software. The Cisco IOS XE DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the router to DHCP clients. The DHCP server can be configured to assign additional parameters such as the IP address of the domain name system (DNS) server and the default router.
DHCP Address Allocation Using Option 82	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.9S	The Cisco IOS XE DHCP server can allocate dynamic IP addresses based on the relay information option (option 82) information sent by the relay agent. The following commands were introduced by this feature: **address range**, **class**, **ip dhcp class**, **ip dhcp use class**, **relay agent information**, **relay-information hex**.
DHCP Statically Configured Routes Using a DHCP Gateway	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.9S	This feature enables the configuration of static routes that point to an assigned DHCP next hop router. The following commands were modified by this feature: **ip route**, **show ip route**.
DHCP Server Options - Import and Autoconfiguration	Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.9S	Options imported by multiple subsystems can co-exist in the DHCP address pool. When the session is terminated or the lease is released, the imported options are cleared.
DHCP Server Multiple Subnet	12.4(15)T 12.2(33)SRB 15.3(1)S Cisco IOS XE Release 3.2SE Cisco IOS XE Release 3.9S	The DHCP Server Multiple Subnet feature enables multiple subnets to be configured under the same DHCP address pool. The following commands were introduced or modified: **network**(DHCP), **override default-router**.

Feature Name	Releases	Feature Configuration Information
DHCP Static Mapping	Cisco IOS XE Release 3.9S	Configuring static mapping pools enables the DHCP server to read the static bindings from a separate text file (similar in format to the DHCP database file) that is stored in special pools. The following commands were introduced or modified: **origin**.
DHCP Server Import All Enhancement	Cisco IOS XE Release 3.9S	The DHCP Server Import All Enhancement feature is an enhancement to the **import all** command. Prior to this feature, the options imported through the **import all** command were overwritten by those imported by another subsystem. Through this feature, options imported by multiple subsystems can coexist in the DHCP address pool. When the session is terminated or the lease is released, the imported options are cleared.
DHCPv4 Client options	Cisco IOS XE Fuji Release 16.9.1	The following features are supported on Cisco 4000 Series ISRs: • Regular Expression support for options 60, 77, 124 and 125 • Generic support to configure all applicable client DHCP options • Import options learnt on specific interface to DHCP pool • Longest Match support for option 60, 77, 124 and 125



CHAPTER 3

Configuring the Cisco IOS XE DHCP Client

Cisco IOS XE Dynamic Host Configuration Protocol (DHCP) client software provides the flexibility to include various configuration options for the DHCP client. A DHCP client is defined as an Internet host using DHCP to obtain configuration parameters such as an IP address. This module describes the concepts and tasks needed to configure the Cisco IOS XE DHCP client.

Feature Information for the Cisco IOS XE DHCP Client

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for the Cisco IOS XE DHCP Client

Feature Name	Releases	Feature Configuration Information
DHCP Client	Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.9S	A DHCP client is defined as an Internet host using DHCP to obtain configuration parameters such as an IP address. The following command was introduced by this feature: **ip address dhcp**

Feature Name	Releases	Feature Configuration Information
Configurable DHCP Client	Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.9S	The configurable DHCP client functionality allows a DHCP client to use a user-specified client identifier, class identifier, or suggested lease time when requesting an address from a DHCP server. The following commands were introduced by this feature: **ip dhcp client class-id**, **ip dhcp client client-id**, **ip dhcp client hostname**, **ip dhcp client lease**, **ip dhcp client request**
DHCPv4 Client Options	Cisco IOS XE Fuji 16.9.1	The DHCP Client supports configuration of all 1-254 options.
DHCP Client Options using unicast mode	Cisco IOS XE Amsterdam 17.2.1	Introduces support for unicast mode on DHCP. This helps with splitting the horizon therefore improving security of the network.

Information About the DHCP Client

DHCP Client Operation

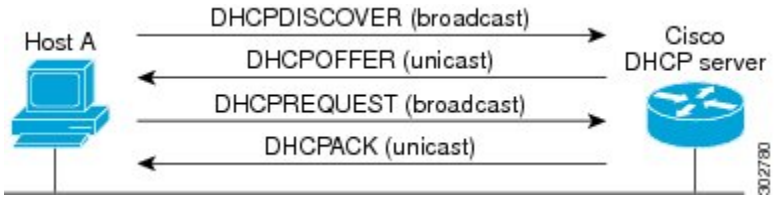
When a Dynamic Host Configuration Protocol (DHCP) client requests an IP address from a DHCP server on a Cisco IOS XE platform, the default process includes:

- DHCPDISCOVERY (broadcast)
- DHCPOFFER (broadcast)
- DHCPREQUEST (broadcast)
- DHCPACK (unicast)

The DHCP on Cisco IOS XE platform supports only broadcast mode with the DHCPOFFER. From Cisco IOS XE Amsterdam Release 17.2, the DHCP on IOS XE platform also supports unicast mode. The DHCP unicast mode helps to split the horizon for security consideration. The DHCP broadcast mode is enabled by default. To enable the DHCP unicast mode, configure the **ip dhcp client broadcast-flag clear** command on the DHCP client. After configuring the command, the DHCPOFFER is sent as a unicast message.

The DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. A DHCP client is an Internet host using DHCP to obtain configuration parameters such as an IP address. The following figure shows the basic steps that occur when a DHCP client requests an IP address from a DHCP server. The client, Host A, sends a DHCPDISCOVER broadcast message to locate a DHCP server. A DHCP server offers configuration parameters (such as an IP address, a MAC address, a domain name, and a lease for the IP address) to the client in a DHCPOFFER unicast/broadcast message.

Figure 3: DHCP Request for an IP Address from a DHCP Server



A DHCP client may receive offers from multiple DHCP servers. However, it can accept any one of the offers; the client usually accepts the first offer it receives. The offer from the DHCP server is not a guarantee that the IP address is allocated to the client. However, the server usually reserves the address until the client has had a chance to formally request the address .

The client returns a formal request for the offered IP address to the DHCP server in a DHCPREQUEST broadcast message. The DHCP server confirms that the IP address is allocated to the client by returning a DHCPACK unicast message to the client.

DHCP Client Overview

The configurable dynamic host configuration protocol client functionality allows a DHCP client to use a user-specified client identifier, class identifier, or suggested lease time when requesting an address from a DHCP server.

Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The DHCP client provides flexibility by allowing the following options to be configured for a DHCP client:

- Option 12—This option specifies the name of the client. The name may or may not be qualified with the local domain.
- Option 51—This option is used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address.
- Option 55—This option allows the DHCP client to request certain options from the DHCP server. The **ip dhcp client request** command allows the system administrator to turn off some of the requested options, thus removing them from the request list.
- Option 60—This option allows the user to configure the vendor class identifier string to use in the DHCP interaction.
- Option 61—This option is used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain.
- Option 77—This option is used by a DHCP clients to optionally identify the type or category of user or applications it represents. The information contained in this option represents the user class of which the client is a member. Based on this class, a DHCP server selects the appropriate address pool to assign an address to the client and the appropriate configuration parameters.
- Option 120—This option is used to specify a 32-bit (binary) IPv4 address to be used by the Session Initiation Protocol (SIP) client to locate a SIP server.

- Option 121—This option is used to configure classless static routes by specifying classless network destinations; that is, each routing table entry includes a subnet mask. Upto ten classless static routes are supported using option 121 on the DHCP client.

Note If a request includes both static routes and classless static routes, the client uses only the classless static routes. If the DHCP server returns both a classless static route option and a router option, the DHCP client ignores the router option.

- Option 124—This option is used by DHCP clients and servers to exchange vendor-class information.
- Option 125—This option is used by DHCP clients and servers to exchange vendor-specific information.

How to Configure the DHCP Client

Configuring the DHCP Client

Cisco devices running Cisco software include the Dynamic Host Configuration Protocol (DHCP) server and relay agent software, which are enabled by default. Your device can act as both the DHCP client and the DHCP server. Use the **ip address dhcp** command to obtain IP address information for the configured interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address dhcp**
5. **end**
6. **debug dhcp detail**
7. **debug ip dhcp server packets**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address dhcp** **Example:** `Device(config-if)# ip address dhcp`	Acquires an IP address on an interface from DHCP.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**debug dhcp detail** **Example:** `Device# debug dhcp detail`	Displays the DHCP packets that were sent and received.
Step 7	**debug ip dhcp server packets** **Example:** `Device# debug ip dhcp server packets`	Displays the server side of the DHCP interaction.

Troubleshooting Tips

To verify the configuration, you can use the **debug dhcp detail** EXEC command to display the DHCP packets that were sent and received. To display the server side of the DHCP interaction, use the **debug ip dhcp server packets** command.

Configure Administrative Distance

To configure the default Dynamic Host Configuration Protocol (DHCP) Administrative Distance (AD), use the **ip dhcp client default-router distance** command in interface configuration or global configuration mode:

Configuration in Interface Configuration Mode:

When you use this command for a interface, AD is applied to the route received in DHCP process of that particular interface.

```
Router # configure terminal
Router(config)# interface FastEthernet 0/2
Router(config-if)# ip dhcp client default-router distance 2
```

Configuration in Global Configuration Mode:

Use this command to configure AD is on the route received in DHCP process of any interface.

```
Router # configure terminal
Router(config)#ip dhcp-client default-router distance 10
```

```
Router(config)#interface e0/0
Router(config-if)#ip address dhcp
Router(config-if)#no shut
Router(config-if)#end
```

To disable the configuration, use the **no** form of this command.

Note When you install the **ip dhcp client default-router distance** command in interface configuration or global configuration mode, default route given by DHCP is installed with specified AD. But, when you use the same command with different AD value, it does not take effect immediately. It takes effect when a new connection is made or when a new Discover-Offer-Request-Ack (DORA) cycle happens. It can be done in any one of the following ways:

- Bounce the interface, which means execute shutdown followed by no shutdown of the interface.
- Release (using **release dhcp** *interface*) command and renew DHCP (using **renew dhcp** *interface*) command from exec mode.

Configuration Examples for the DHCP Client

Configuring the DHCP Client Example

The figure below shows a simple network diagram of a DHCP client on an Ethernet LAN.

Figure 4: Topology Showing DHCP Client with GigabitEthernet Interface

On the DHCP server, the configuration is as follows:

```
ip dhcp pool 1
 network 10.1.1.0 255.255.255.0
 lease 1 6
```

On the DHCP client, the configuration is as follows on interface GigabitEthernet 0/0/0:

```
interface GigabitEthernet 0/0/0
 ip address dhcp
```

This configuration allows the DHCP client to acquire an IP address from the DHCP server through GigabitEthernet interface 0/0/0.

Customizing the DHCP Client Configuration Example

The following example shows how to customize the DHCP client configuration with various options on GigabitEthernet interface 0/0/1:

```
interface GigabitEthernet 0/0/1
 ip dhcp client client-id ascii my-test1
 ip dhcp client class-id my-class-id
 ip dhcp client lease 0 1 0
 ip dhcp client hostname sanfran
 no ip dhcp client request tftp-server-address
 ip address dhcp
```

The following example shows DHCP Client configuration on GigabitEthernet 0/0/1 to generically request options:

```
!
interface GigabitEthernet 0/0/1
ip dhcp client request option 4 5 7 8 9 10 11 17 18 40 41 42 66 68 69 70 71 72 73 74 75 76
 124 138 141 142 160
no ip address
shutdown
!
```

The following example shows how to configure DHCP Client options with parameters, IP address and string:

```
!
interface GigabitEthernet 0/0/1
ip dhcp client option 1 ip 10.0.0.1
ip dhcp client option 13 ascii test13
ip dhcp client option 14 ascii test14
ip dhcp client option 16 ip 10.0.0.16
ip dhcp client option 46 ascii test46
ip dhcp client option 47 ascii test47
ip dhcp client option 50 ip 10.0.0.50
ip dhcp client option 51 ascii test51
ip dhcp client option 52 ascii test52
ip dhcp client option 54 ascii test54
ip dhcp client option 58 ascii test58
ip dhcp client option 59 ascii test59
ip dhcp client option 60 ascii test60
ip dhcp client option 61 ascii test61
ip dhcp client option 62 ascii test62
ip dhcp client option 63 ip 10.0.0.63
ip dhcp client option 64 ascii test64
ip dhcp client option 65 ip 10.0.0.65
ip dhcp client option 67 ascii test67
ip dhcp client option 90 ascii test90
ip dhcp client option 116 ascii test116
ip dhcp client option 118 ip 10.0.0.118
ip dhcp client option 220 ip 10.0.0.220
ip dhcp client option 221 ascii test221
ip address dhcp
shutdown
!
```

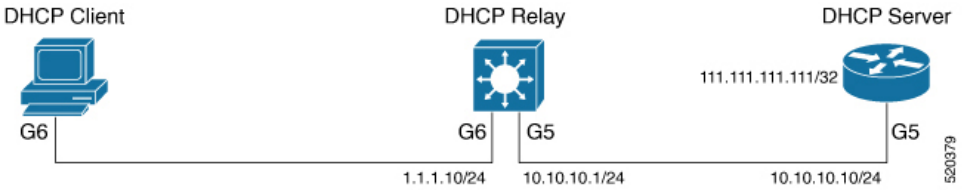
The following example shows how to configure DHCP Client options with class::

Example: Configuring the DHCP Client in Unicast Mode

The following example shows how to configure DHCP Client in unicast mode:

The figure below shows a simple network diagram of a DHCP client in unicast mode.

Figure 5: Topology Showing DHCP Client with GigabitEthernet Interface



Client:

```
interface GigabitEthernet6
ip address dhcp
ip dhcp client broadcast-flag clear
```

Relay:

```
interface GigabitEthernet6
ip address 1.1.1.10 255.255.255.0
ip helper-address 111.111.111.111

!
interface GigabitEthernet5
  ip address 10.10.10.1 255.255.255.0

ip route 111.111.111.111 255.255.255.255 GigabitEthernet5
```

Server:

```
interface Loopback10
ip address 111.111.111.111 255.255.255.255
no shutdown
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet5

ip dhcp pool Cisco
network 11.11.11.0 255.255.255.0
default-router 11.11.11.1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples.	Cisco IOS IP Addressing Services Command Reference
DHCP conceptual information	“DHCP Overview” module

Related Topic	Document Title
DHCP server configuration	"Configuring the Cisco IOS XE DHCP Server" module
DHCP server on-demand address pools	"Configuring the DHCP Server On-Demand Address Pool Manager" module
DHCP relay agent configuration	"Configuring the Cisco IOS XE DHCP Relay Agent" module
DHCP advanced features	"Configuring DHCP Services for Accounting and Security" module

RFCs

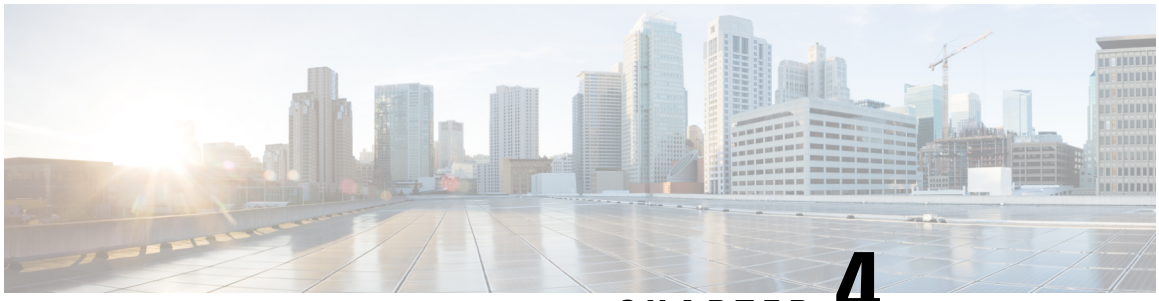
RFCs	Title
RFC 2131	Dynamic Host Configuration Protocol
RFC 2132	DHCP Options and BOOTP Vendor Extensions

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport



CHAPTER 4

Implementing DHCP for IPv6

This module describes how to configure Dynamic Host Configuration Protocol (DHCP) for IPv6.

DHCPv6 Prefix Delegation

The IPv6 Access Services—DHCPv6 Prefix Delegation feature can be used to manage link, subnet, and site addressing changes. Dynamic Host Configuration Protocol for IPv6 (DHCPv6) can be used in environments to deliver stateful and stateless information, which are defined as follows:

- Stateful prefix delegation—Address assignment is centrally managed and clients must obtain configuration information such as address autoconfiguration and neighbor discovery that is not available through protocols.
- Stateless prefix delegation—Stateless configuration parameters do not require a server to maintain any dynamic state for individual clients, such as Domain Name System (DNS) server addresses and domain search list options.

Extensions to DHCPv6 also enable prefix delegation, through which an ISP can automate the process of assigning prefixes to a customer for use within the customer's network. The prefix delegation occurs between a provider edge (PE) device and customer premises equipment (CPE) using the DHCPv6 prefix delegation option. Once the ISP has delegated prefixes to a customer, the customer may further subnet and assign prefixes to the links in the customer's network.

Configuring Nodes Without Prefix Delegation

Stateless DHCPv6 allows DHCPv6 to be used for configuring a node with parameters that do not require a server to maintain any dynamic state for the node. The use of stateless DHCP is controlled by router advertisement (RA) messages multicasted by routers. The Cisco IOS XE DHCPv6 client will invoke stateless DHCPv6 when it receives an RA. The Cisco IOS XE DHCPv6 server will respond to a stateless DHCPv6 request with configuration parameters, such as the DNS servers and domain search list options.

Client and Server Identification

Each DHCPv6 client and server is identified by a DHCP unique identifier (DUID). The DUID is carried in client identifier and server identifier options. The DUID is unique across all DHCP clients and servers, and it is stable for any specific client or server. DHCPv6 uses DUIDs based on link-layer addresses for both the client and server identifier. The device uses the MAC address from the lowest-numbered interface to form the DUID. The network interface is assumed to be permanently attached to the device.

When a DHCPv6 client requests two prefixes with the same DUID but with different identity association identifiers (IAIDs) on two different interfaces, these prefixes are considered to be for two different clients, and the interface information is maintained for both.

Rapid Commit

The DHCPv6 client can obtain configuration parameters from a server either through a rapid two-message exchange (solicit, reply) or through a four-message exchange (solicit, advertise, request, and reply). By default, the four-message exchange is used. When the rapid-commit option is enabled by both the client and the server, the two-message exchange is used.

DHCPv6 Client and Relay Functions

The DHCPv6 client and relay functions are mutually exclusive on an interface. When one of these functions is enabled and a user tries to configure a different function on the same interface, one of the following messages is displayed: “Interface is in DHCP client mode” or “Interface is in DHCP relay mode.”

The following sections describe these functions:

Client Function

The DHCPv6 client function can be enabled on individual IPv6-enabled interfaces.

Note You need APPX license package to enable the DHCPv6 client function on the device.

The DHCPv6 client can request and accept those configuration parameters that do not require a server to maintain any dynamic state for individual clients, such as DNS server addresses and domain search list options.

The DHCPv6 client can also request the delegation of prefixes. The prefixes acquired from a delegating device will be stored in a local IPv6 general prefix pool. The prefixes in the general prefix pool can then be referred to from other applications; for example, the general prefix pool can be used to number device downstream interfaces.

For IPv6, the details of the default gateway are received from router advertisement (RA) and not from the DHCP server. The details of the default gateway obtained from the DHCP server are not added in the client server. Hence, the following IPv6 commands must be configured on the client interface:

- **pv6 nd autoconfig prefix”**
- **“ipv6 nd autoconfig default”**

Server Selection

A DHCPv6 client builds a list of potential servers by sending a solicit message and by collecting advertise message replies from servers. These messages are ranked based on the preference value, and servers may add a preference option to their advertise messages explicitly stating their preference value. If the client needs to acquire prefixes from servers, only servers that have advertised prefixes are considered.

IAPD and IAID

An Identity Association for Prefix Delegation (IAPD) is a collection of prefixes assigned to a requesting device. A requesting device may have more than one IAPD; for example, one for each of its interfaces.

Each IAPD is identified by an IAID. The IAID is chosen by the requesting device and is unique among the IAPD IAIDs on the requesting device. IAIDs are made consistent across reboots by using information from the associated network interface, which is assumed to be permanently attached to the device.

DHCPv6 Relay Agent

A DHCPv6 relay agent, which may reside on the client's link, is used to relay messages between the client and the server. The DHCPv6 relay agent operation is transparent to the client. A DHCPv6 client locates a DHCPv6 server using a reserved, link-scoped multicast address. For direct communication between the DHCPv6 client and the DHCPv6 server, both of them must be attached to the same link. However, in some situations where ease of management, economy, or scalability is a concern, it is desirable to allow a DHCPv6 client to send a message to a DHCPv6 server that is not connected to the same link. IPv6 enable is required for IPv6 DHCP relay, although IPv6 address is configured.

DHCPv6 Relay Agent Notification for Prefix Delegation

The DHCPv6 relay agent notification for prefix delegation allows the device working as a DHCPv6 relay agent to find prefix delegation options by reviewing the contents of a DHCPv6 RELAY-REPLY packet that is relayed by the relay agent to the client. When a prefix delegation option is found by the relay agent, the relay agent extracts the information about the prefix that is being delegated and inserts an IPv6 static route matching the prefix delegation information onto the relay agent. Future packets destined to that prefix via relay will be forwarded based on the information contained in the prefix delegation. The IPv6 static route is then left in the routing table until the prefix delegation lease time expires or the relay agent receives a release packet from the client releasing the prefix delegation.

No user configuration is required for this feature. Static route management is done automatically by the relay agent.

IPv6 routes are added when the relay agent relays a RELAY-REPLY packet, and IPv6 routes are deleted when the prefix delegation lease time expires or the relay agent receives a release message. An IPv6 static route in the routing table of the relay agent can be updated when the prefix delegation lease time is extended.

The DHCP—DHCPv6 Relay Agent Notification for Prefix Delegation feature leaves a static IPv6 route on the routing table of the relay agent. The registered IPv6 address allows unicast reverse packet forwarding (uRPF) to work by allowing the device doing the reverse lookup to confirm that the IPv6 address on the relay agent is not malformed or spoofed. The static route that remains in the routing table of the relay agent can be redistributed to other routing protocols to advertise the subnets to other nodes. Static routes will be removed when a DHCP_DECLINE message is sent by the client.

DHCPv6 Relay Options: Remote-ID for Gigabit Ethernet and Fast Ethernet Interfaces

The DHCPv6 Ethernet Remote ID Option feature adds the remote identification (remote-ID) option to relayed (RELAY-FORWARD) DHCPv6 packets.

The remote-ID option provides information to the DHCPv6 server, which includes port information, the system's DUID, and the VLAN ID. This information can be used to uniquely identify both the relay and the port on the relay through which the client packet arrived. The DHCPv6 server uses this information to select parameters specific to a particular user, host, or subscriber modem.

The addition of the remote-ID option to the RELAY-FORWARD packet occurs automatically and no user configuration is necessary.

The DHCPv6 server does not need to echo the remote-ID option in the RELAY-REPLY packet. The Internet Assigned Numbers Authority (IANA) has assigned the DHCPv6 option code 37 for the relay agent remote-ID option.

If the remote-ID option is included in the RELAY-REPLY packet, the option is removed from the packet before it is relayed to the client.

DHCPv6 Relay Options: Reload Persistent Interface ID

The DHCPv6 Relay—Reload Persistent Interface ID Option feature makes the interface ID option persistent. The interface ID is used by relay agents to decide which interface should be used to forward a RELAY-REPLY packet. A persistent interface-ID option will not change if the device acting as a relay agent goes offline during a reload or a power outage. When the device acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as, when the relay agent reboots and the number of interfaces in the interface index changes, or when the relay agents boot up and has more virtual interfaces than it did before the reboot). This feature prevents such scenarios from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as the short form of the interface name. The interface name as the DHCPv6 interface ID helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

DHCPv6 Relay Chaining

DHCPv6 messages can be relayed through multiple relay agents. This configuration is called *relay chaining*. A relay chaining configuration can be supported only when each relay agent adds information to DHCPv6 messages before relaying them. The information helps in relaying the DHCPv6 reply back to the DHCPv6 client through the same path.

The delegated IPv6 prefix must be routable in order to be useful. The actual DHCPv6 Prefix Delegation (PD) client may not be permitted to inject routes into the delegating network. In service provider (SP) networks, for example, an edge device typically acts as a DHCPv6 relay agent, and this edge device often has the responsibility to maintain routes within the SP network for clients' PD bindings. In the event that DHCPv6 requests and responses are relayed through a chain of DHCPv6 relays, there may be a need to introduce appropriate routes (particularly with DHCPv6 PD) in the Forwarding Information Base (FIB) so that routing is handled transparently.

DHCPv6 Relay SSO and ISSU

In specific Cisco networking devices that support dual route processors (RPs), stateful switchover (SSO) takes advantage of RP redundancy to increase network availability. The feature establishes one of the RPs as the active processor while the other RP is designated as the standby processor, and then synchronizing critical state information between them. Following an initial synchronization between the two processors, SSO dynamically maintains RP state information between them.

The Cisco IOS XE In Service Software Upgrade (ISSU) process allows the Cisco IOS XE software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades

are a significant cause of downtime. The ISSU allows the Cisco IOS XE software to be modified while packet forwarding continues. This increases network availability and reduces downtime caused by planned software upgrades.

The SSO and the ISSU use redundant hardware, with the active and standby RP each running an instance of the DHCPv6 relay agent. Both instances exchange run-time state data.

For further information about the SSO and the ISSU, see the "Stateful Switchover" and the "Cisco IOS XE In Service Software Upgrade" modules respectively, in the Cisco IOS High Availability Configuration Guide.

DHCPv6 Relay Options: Remote-ID for Ethernet Interfaces

This feature adds the remote identification (remote-ID) option to relayed (RELAY-FORWARD) DHCPv6 packets.

The remote-ID option provides information to the DHCPv6 server, including port information, the system's DUID, and the VLAN ID. Collectively, this information can be used to uniquely identify both the relay and the port on the relay through which the client's packet arrived. The DHCPv6 server uses this information to select parameters specific to a particular user, host, or subscriber modem. This feature works only for Ethernet interfaces at this time.

This feature introduces no user configuration. Because the addition of the remote-ID option to the RELAY-FORWARD packet occurs automatically, no user configuration is necessary.

The DHCPv6 server does not need to echo the remote-ID option in the RELAY-REPLY packet. Internet Assigned Numbers Authority (IANA) has assigned the DHCPv6 option code 37 for the relay agent remote-ID option.

If the remote-ID option is included in the RELAY-REPLY packet, the option is stripped out of the packet before the packet is relayed to the client.

DHCPv6 Relay Options: Reload Persistent Interface-ID

This feature makes the interface-ID option, which is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet, persistent. A persistent interface-ID option will not change if the router acting as a relay agent goes offline (such as during a reload or a power outage). When the router acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as cases where the relay agent reboots and has a change in the number of interfaces in the interface index, or the relay agents boots up and has more virtual interfaces than it did before the reboot). This feature prevents this scenario from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as simply the short form of the interface name. This syntax helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

How to Implement DHCP for IPv6

Configuring the DHCPv6 Server Function

Configuring the DHCPv6 Configuration Pool

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **domain-name** *domain*
5. **dns-server** *ipv6-address*
6. **prefix-delegation** *ipv6-prefix* / *prefix-length* *client-duid* [**iaid** *iaid*] [*lifetime*]
7. **prefix-delegation pool** *poolname* [**lifetime** *valid-lifetime preferred-lifetime*]
8. **exit**
9. **interface** *type* *number*
10. **ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool pool1`	Configures a DHCPv6 configuration information pool and enters DHCPv6 pool configuration mode. The *pool name* can be a string, such as "abcd" or an integer value, such as 0. During execution, the configuration mode changes to DHCPv6 pool configuration mode. In this mode, you can configure pool parameters, such as prefixes to be delegated, DNS servers, and so on.
Step 4	**domain-name** *domain* **Example:** `Device(config-dhcp)# domain-name example.com`	Configures a domain name for a DHCPv6 client.

	Command or Action	Purpose
Step 5	**dns-server** *ipv6-address* **Example:** `Device(config-dhcp)# dns-server 2001:DB8:3000:3000::42`	Specifies the DNS IPv6 servers available to a DHCPv6 client.
Step 6	**prefix-delegation** *ipv6-prefix* / *prefix-length client-duid* [**iaid** *iaid*] [*lifetime*] **Example:** `Device(config-dhcp)# prefix-delegation 2001:DB8:1263::/48 0005000400F1A4D070D03`	Specifies a manually configured numeric prefix to be delegated to a specified client's IAPD.
Step 7	**prefix-delegation pool** *poolname* [**lifetime** *valid-lifetime preferred-lifetime*] **Example:** `Device(config-dhcp)# prefix-delegation pool pool1 lifetime 1800 60`	Specifies a named IPv6 local prefix pool from which prefixes are delegated to DHCPv6 clients. The lifetime values are *valid-lifetime* and *preferred-lifetime*. These are referred to as T1 and T2. When the T2 expires, a renew request is sent to the particular server and if the client does not get a response within T1, the client sends a REBIND request to all available servers. The value of **lifetime** can be specified as: • a fixed duration that remains constant across consecutive advertisements • absolute expiration time in the future, so that the advertised lifetime decrements in real time and is equal to zero at the specified time. The specified duration is between 60 and 4294967295 seconds or infinity if the keyword **infinite** is specified. If the lifetimes are not specified, by default, the *valid-lifetime* is 2592000 seconds (or 30 days) and the *preferred-lifetime* is 604800 seconds (or 7 days).
Step 8	**exit** **Example:** `Device(config-dhcp)# exit`	Exits DHCPv6 pool configuration mode, and returns the device to global configuration mode.
Step 9	**interface** *type number* **Example:** `Device(config)# interface serial 3`	Specifies an interface type and number, and enters interface configuration mode.
Step 10	**ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**]	Enables or disables DHCPv6 service on an interface.

	Command or Action	Purpose
	Example: `Device(config-if)# ipv6 dhcp server pool1`	• If specified, **rapid-commit** enables the use of the two-message exchange for prefix delegation and other configuration. If the **rapid-commit** option is used in the Solicit message and **rapid-commit** is enabled for the server, the server responds to the Solicit with a Reply message. By default, **rapid-commit** is disabled. • Default value of **preference** is 0. • If the **allow-hint** option is specified, is a valid prefix in the associated local prefix pool and is not assigned to anybody, the server delegates the client-suggested prefix in the Solicit and Request messages. Otherwise, the hint is ignored and a prefix is delegated from the free list in the pool. **Note** By default, DHCPv6 service on an interface is disabled.
Step 11	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring a Binding Database Agent for the Server Function

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp database** *agent* [**write-delay** *seconds*] [**timeout** *seconds*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp database** *agent* [**write-delay** *seconds*] [**timeout** *seconds*]	Specifies DHCPv6 binding database agent parameters.

	Command or Action	Purpose
	Example: `Device(config)# ipv6 dhcp database tftp://10.0.0.1/dhcp-binding`	• • **agent-URL**—flash, NVRAM, FTP, TFTP, or RCP uniform resource locator. • **write-delay**—specifies how often DHCP sends database updates. By default, DHCPv6 server waits 300 seconds before transmitting database changes. The minimum delay is 60 seconds. • **timeout**—specifies the time to wait for a database transfer. Infinity is defined as zero seconds. Transfers that exceed the timeout period are aborted. Default value is 300 seconds.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring the DHCPv6 Client Function

General prefixes can be defined dynamically from a prefix received by a DHCPv6 prefix delegation client. The delegated prefix is stored in a general prefix.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 dhcp client pd** {*prefix-name* | **hint** *ipv6-prefix*} [**rapid-commit**]
5. **ipv6 nd autoconfig prefix**
6. **ipv6 nd autoconfig default-router**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ipv6 dhcp client pd** {*prefix-name* | **hint** *ipv6-prefix*} [**rapid-commit**] **Example:** `Device(config-if)# ipv6 dhcp client pd dhcp-prefix`	Enables the DHCPv6 client process and enables a request for prefix delegation through a specified interface. **Note** The **ipv6 dhcp client pd hint** command should always be used along with the **ipv6 dhcp client pd** *pd-name* command.
Step 5	**ipv6 nd autoconfig prefix** **Example:** `Device(config-if)# ipv6 nd autoconfig prefix`	Allows Neighbor Discovery to install all valid on-link prefixes from router advertisements (RAs) received on the interface.
Step 6	**ipv6 nd autoconfig default-router** **Example:** `Device(config-if)# ipv6 nd autoconfig default-router`	Allows Neighbor Discovery to install a default route to the Neighbor Discovery-derived default router.
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Example

The following example depicts the usage of **ipv6 dhcp client pd hint** command along with the **ipv6 dhcp client pd** *pd-name* command.

```
interface GigabitEthernet0/10
no ip address
media-type auto-select
negotiation auto
ipv6 address prefix-from-provider ::1/48
ipv6 enable
ipv6 dhcp client pd hint 2001:DB8:43::/48
ipv6 dhcp client pd prefix-from-provider
```

Configuring the DHCPv6 Relay Agent

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **ipv6 enable**
5. **ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 4/2/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 5	**ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number*] **Example:** `Device(config-if) ipv6 dhcp relay destination FE80::250:A2FF:FEBF:A056 gigabitethernet 4/3/0`	Specifies a destination address to which client packets are forwarded and enables the DHCPv6 relay service on the interface.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring Route Addition for Relay and Server

To enable route addition by DHCPv6 relay and server for the delegated prefix, use the **ipv6 dhcp iapd-route-add** command in global configuration mode.

DHCPv6 relay inserts a route for the delegated prefix without additional configuration (i.e., the default is **ipv6 dhcp iapd-route-add** , which of course isn't NVGEN'ed.) If you want to disable this insertion, you must configure **no ipv6 dhcp iapd-route-add** .

The relay tracks valid and preferred lifetimes for the delegated prefix. When the prefix reaches the end of the valid lifetime, the route is automatically removed from the routing table.

To add routes for individually assigned IPv6 addresses on the relay or server, use the **ipv6 dhcp iana-route-add** command in global configuration mode.

Defining a General Prefix with the DHCPv6 Prefix Delegation Client Function

Perform this task to configure the DHCPv6 client function on an interface and enable prefix delegation on an interface. The delegated prefix is stored in a general prefix.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 dhcp client pd** {*prefix-name* | **hint** *ipv6-prefix*} [**rapid-commit**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ipv6 dhcp client pd** {*prefix-name* | **hint** *ipv6-prefix*} [**rapid-commit**] **Example:** `Device(config-if)# ipv6 dhcp client pd dhcp-prefix`	Enables the DHCPv6 client process and enables a request for prefix delegation through a specified interface. • The delegated prefix is stored in the general prefix *prefix-name* argument.

Configuring a VRF-Aware Relay for MPLS VPN Support

Configuring a VRF-Aware Relay

Note You do not have to configure this feature on specified interfaces. If you want the feature to be enabled globally only on a device, perform steps 1, 2, and 3

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp-relay option vpn**
4. **interface** *type number*
5. **ipv6 dhcp relay option vpn**
6. **ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number* | **vrf** *vrf-name* | **global**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp-relay option vpn** **Example:** `Device(config)# ipv6 dhcp-relay option vpn`	Enables the DHCP for IPv6 relay VRF-aware feature globally.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 5	**ipv6 dhcp relay option vpn** **Example:** `Device(config-if)# ipv6 dhcp relay option vpn`	Enables the DHCP for IPv6 relay VRF-aware feature on the specified interface. Enabling this command supersedes the configuration that is enabled by using the **ipv6 dhcp-relay option vpn** command.
Step 6	**ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number*	**vrf** *vrf-name*	**global**] **Example:** `Device(config-if)# ipv6 dhcp relay destination FE80::250:A2FF:FEBF:A056 ethernet 0/0`	Specifies a destination address to which client messages are forwarded.
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Restarting the DHCPv6 Client on an Interface

Perform this task to restart the DHCPv6 client on a specified interface after first releasing and unconfiguring previously acquired prefixes and other configuration options.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 dhcp client** *interface-type interface-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 dhcp client** *interface-type interface-number* **Example:** `Device# clear ipv6 dhcp client GigabitEthernet 1/0/0`	Restarts the DHCPv6 client on an interface.

Deleting Automatic Client Bindings from the DHCPv6 Binding Table

SUMMARY STEPS

1. **enable**
2. **clear ipv6 dhcp binding** [*ipv6-address*] [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 dhcp binding** [*ipv6-address*] [**vrf** *vrf-name*] **Example:** `Device# clear ipv6 dhcp binding`	Deletes automatic client bindings from the DHCPv6 binding table.

Troubleshooting DHCPv6

Note Step 1 is common to each debug command. Step 2 to Step 5 are separate debugging commands that can be used in any order.

SUMMARY STEPS

1. **enable**
2. **debug ipv6 dhcp** [**detail**]
3. **debug ipv6 dhcp database**
4. **debug ipv6 dhcp relay**
5. **debug ipv6 dhcp redundancy** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ipv6 dhcp** [**detail**] **Example:** `Device# debug ipv6 dhcp`	Enables debugging for DHCPv6.
Step 3	**debug ipv6 dhcp database** **Example:** `Device# debug ipv6 dhcp database`	Enables debugging for the DHCPv6 binding database.
Step 4	**debug ipv6 dhcp relay** **Example:** `Device# debug ipv6 dhcp relay`	Enables DHCPv6 relay agent debugging.
Step 5	**debug ipv6 dhcp redundancy** [**detail**] **Example:** `Device# debug ipv6 dhcp redundancy`	Enables DHCPv6 redundancy debugging.

Verifying the DHCPv6 Configuration

SUMMARY STEPS

1. **enable**
2. **show ipv6 dhcp**
3. **show ipv6 dhcp binding** [*ipv6-address*]

4. **show ipv6 dhcp database** [*agent-URL*]
5. **show ipv6 dhcp interface** [*type number*]
6. **show ipv6 dhcp pool** [*poolname*]
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 dhcp** **Example:** `Device# show ipv6 dhcp`	Displays the DUID on a specified device.
Step 3	**show ipv6 dhcp binding** [*ipv6-address*] **Example:** `Device# show ipv6 dhcp binding`	Displays automatic client bindings from the DHCPv6 database.
Step 4	**show ipv6 dhcp database** [*agent-URL*] **Example:** `Device# show ipv6 dhcp database`	Displays the DHCPv6 binding database agent information.
Step 5	**show ipv6 dhcp interface** [*type number*] **Example:** `Device# show ipv6 dhcp interface`	Displays DHCPv6 interface information.
Step 6	**show ipv6 dhcp pool** [*poolname*] **Example:** `Device# show ipv6 dhcp pool`	Displays DHCPv6 configuration pool information.
Step 7	**show running-config** **Example:** `Device# show running-config`	Displays the current configuration running on the router.

Example Verifying the DHCPv6 Configuration

Sample Output from the show ipv6 dhcp Command

The following sample output from the **show ipv6 dhcp** command displays the DUID of the device:

```
Device# show ipv6 dhcp

This device's DHCPv6 unique identifier(DUID): 000300010002FCA5DC1C
```

Sample Output from the show ipv6 dhcp binding Command

In the following sample output, the **show ipv6 dhcp binding** command displays information about two clients, including their DUIDs, IAPDs, prefixes, and preferred and valid lifetimes:

```
Device# show ipv6 dhcp binding

Client: FE80::202:FCFF:FEA5:DC39 (GigabitEthernet2/1/0)
DUID: 000300010002FCA5DC1C
IA PD: IA ID 0x00040001, T1 0, T2 0
Prefix: 3FFE:C00:C18:11::/68
preferred lifetime 180, valid lifetime 12345
expires at Nov 08 2002 02:24 PM (12320 seconds)
Client: FE80::202:FCFF:FEA5:C039 (GigabitEthernet2/1/0)
DUID: 000300010002FCA5C01C
IA PD: IA ID 0x00040001, T1 0, T2 0
Prefix: 3FFE:C00:C18:1::/72
preferred lifetime 240, valid lifetime 54321
expires at Nov 09 2002 02:02 AM (54246 seconds)
Prefix: 3FFE:C00:C18:2::/72
preferred lifetime 300, valid lifetime 54333
expires at Nov 09 2002 02:03 AM (54258 seconds)
Prefix: 3FFE:C00:C18:3::/72
preferred lifetime 280, valid lifetime 51111
```

Sample Output from the show ipv6 dhcp database Command

The following sample output from the **show ipv6 dhcp database** command shows information on the binding database agents TFTP, NVRAM, and flash:

```
Device# show ipv6 dhcp database

Database agent tftp://172.19.216.133/db.tftp:
write delay: 69 seconds, transfer timeout: 300 seconds
last written at Jan 09 2003 01:54 PM,
write timer expires in 56 seconds
last read at Jan 06 2003 05:41 PM
successful read times 1
failed read times 0
successful write times 3172
failed write times 2
Database agent nvram:/dhcpv6-binding:
write delay: 60 seconds, transfer timeout: 300 seconds
last written at Jan 09 2003 01:54 PM,
write timer expires in 37 seconds
last read at never
successful read times 0
failed read times 0
successful write times 3325
failed write times 0
Database agent flash:/dhcpv6-db:
write delay: 82 seconds, transfer timeout: 3 seconds
last written at Jan 09 2003 01:54 PM,
write timer expires in 50 seconds
last read at never
successful read times 0
failed read times 0
successful write times 2220
failed write times 614
```

Sample Output from the show ipv6 dhcp interface Command

The following is sample output from the **show ipv6 dhcp interface** command. In the first example, the command is used on a device that has an interface acting as a DHCPv6 server. In the second example, the command is used on a router that has an interface acting as a DHCPv6 client:

```
Device# show ipv6 dhcp interface

GigabitEthernet2/1/0 is in server mode
Using pool: svr-p1
Preference value: 20
Rapid-Commit is disabled
Router2# show ipv6 dhcp interface
GigabitEthernet2/1/0 is in client mode
State is OPEN (1)
List of known servers:
Address: FE80::202:FCFF:FEA1:7439, DUID 000300010002FCA17400
Preference: 20
IA PD: IA ID 0x00040001, T1 120, T2 192
Prefix: 3FFE:C00:C18:1::/72
preferred lifetime 240, valid lifetime 54321
expires at Nov 08 2002 09:10 AM (54319 seconds)
Prefix: 3FFE:C00:C18:2::/72
preferred lifetime 300, valid lifetime 54333
expires at Nov 08 2002 09:11 AM (54331 seconds)
Prefix: 3FFE:C00:C18:3::/72
preferred lifetime 280, valid lifetime 51111
expires at Nov 08 2002 08:17 AM (51109 seconds)
DNS server: 2001:DB8:1001::1
DNS server: 2001:DB8:1001::2
Domain name: example1.net
Domain name: example2.net
Domain name: example3.net
Prefix name is cli-p1
Rapid-Commit is enabled
```

Sample Output from the show ipv6 dhcp pool Command

In the following sample output, the **show ipv6 dhcp pool** command displays information about the configuration pool named svr-p1, including static bindings, prefix information, the DNS server, and the domain names found in the svr-p1 pool:

```
Device# show ipv6 dhcp pool

DHCPv6 pool: svr-p1
Static bindings:
Binding for client 000300010002FCA5C01C
IA PD: IA ID 00040002,
Prefix: 3FFE:C00:C18:3::/72
preferred lifetime 604800, valid lifetime 2592000
IA PD: IA ID not specified; being used by 00040001
Prefix: 3FFE:C00:C18:1::/72
preferred lifetime 240, valid lifetime 54321
Prefix: 3FFE:C00:C18:2::/72
preferred lifetime 300, valid lifetime 54333
Prefix: 3FFE:C00:C18:3::/72
preferred lifetime 280, valid lifetime 51111
Prefix from pool: local-p1, Valid lifetime 12345, Preferred lifetime 180
DNS server: 2001:DB8:1001::1
DNS server: 2001:DB8:1001::2
Domain name: example1.net
Domain name: example2.net
Domain name: example3.net
```

```
Active clients: 2
Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2009
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2009 by name01
!
hostname Router
!
ip cef
ipv6 unicast-routing
ipv6 cef
ipv6 cef accounting prefix-length
!
!
interface GigabitEthernet0/0/0
ip address 10.4.9.11 255.0.0.0
media-type 10BaseT
ipv6 address 2001:DB8:C18:1::/64 eui-64
```

Configuration Examples for Implementing DHCPv6

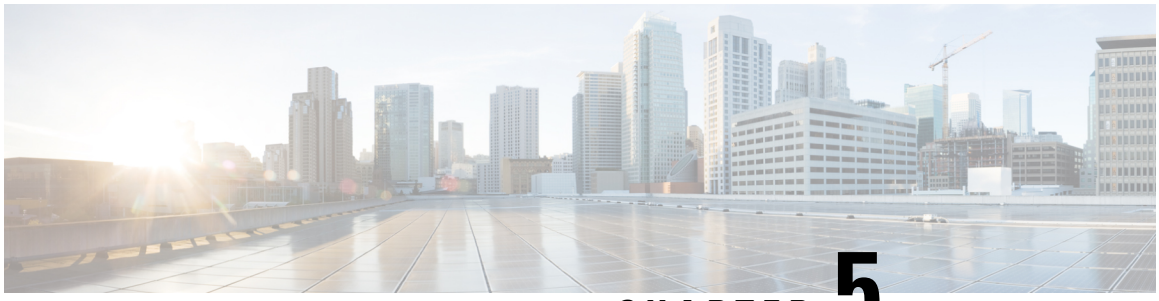
Example: Configuring the DHCPv6 Client Function

In the following example, this Dynamic Host Configuration Protocol for IPv6 (DHCPv6) client has three interfaces. Ethernet interface 0/0 is the upstream link to a service provider, which has a DHCPv6 server function enabled. The Fast Ethernet interfaces 0/0 and 0/1 are links to local networks.

The upstream interface, Ethernet interface 0/0, has the DHCPv6 client function enabled. Prefixes delegated by the provider are stored in the general prefix called prefix-from-provider.

The local networks, Fast Ethernet interfaces 0/0 and 0/1, both assign interface addresses based on the general prefix called prefix-from-provider. The bits on the left of the addresses come from the general prefix, and the bits on the right of the addresses are specified statically.

```
interface Ethernet 0/0
 description uplink to provider DHCP IPv6 server
 ipv6 dhcp client pd prefix-from-provider
!
interface FastEthernet 0/0
 description local network 0
 ipv6 address prefix-from-provider ::5:0:0:0:100/64
!
interface FastEthernet 0/1
 description local network 1
 ipv6 address prefix-from-provider ::6:0:0:0:100/64
```



CHAPTER 5

IPv6 Access Services: DHCPv6 Relay Agent

A Dynamic Host Configuration Protocol for IPv6 (DHCPv6) relay agent, which may reside on the client's link, is used to relay messages between the client and the server.

Information About IPv6 Access Services: DHCPv6 Relay Agent

DHCPv6 Relay Agent

A DHCPv6 relay agent, which may reside on the client's link, is used to relay messages between the client and the server. The DHCPv6 relay agent operation is transparent to the client. A DHCPv6 client locates a DHCPv6 server using a reserved, link-scoped multicast address. For direct communication between the DHCPv6 client and the DHCPv6 server, both of them must be attached to the same link. However, in some situations where ease of management, economy, or scalability is a concern, it is desirable to allow a DHCPv6 client to send a message to a DHCPv6 server that is not connected to the same link. IPv6 enable is required for IPv6 DHCP relay, although IPv6 address is configured.

DHCPv6 Relay Agent Notification for Prefix Delegation

The DHCPv6 relay agent notification for prefix delegation allows the device working as a DHCPv6 relay agent to find prefix delegation options by reviewing the contents of a DHCPv6 RELAY-REPLY packet that is relayed by the relay agent to the client. When a prefix delegation option is found by the relay agent, the relay agent extracts the information about the prefix that is being delegated and inserts an IPv6 static route matching the prefix delegation information onto the relay agent. Future packets destined to that prefix via relay will be forwarded based on the information contained in the prefix delegation. The IPv6 static route is then left in the routing table until the prefix delegation lease time expires or the relay agent receives a release packet from the client releasing the prefix delegation.

No user configuration is required for this feature. Static route management is done automatically by the relay agent.

IPv6 routes are added when the relay agent relays a RELAY-REPLY packet, and IPv6 routes are deleted when the prefix delegation lease time expires or the relay agent receives a release message. An IPv6 static route in the routing table of the relay agent can be updated when the prefix delegation lease time is extended.

The DHCP—DHCPv6 Relay Agent Notification for Prefix Delegation feature leaves a static IPv6 route on the routing table of the relay agent. The registered IPv6 address allows unicast reverse packet forwarding (uRPF) to work by allowing the device doing the reverse lookup to confirm that the IPv6 address on the relay agent is not malformed or spoofed. The static route that remains in the routing table of the relay agent can be redistributed to other routing protocols to advertise the subnets to other nodes. Static routes will be removed when a DHCP_DECLINE message is sent by the client.

DHCPv6 Relay Options: Remote-ID for Gigabit Ethernet and Fast Ethernet Interfaces

The DHCPv6 Ethernet Remote ID Option feature adds the remote identification (remote-ID) option to relayed (RELAY-FORWARD) DHCPv6 packets.

The remote-ID option provides information to the DHCPv6 server, which includes port information, the system's DUID, and the VLAN ID. This information can be used to uniquely identify both the relay and the port on the relay through which the client packet arrived. The DHCPv6 server uses this information to select parameters specific to a particular user, host, or subscriber modem.

The addition of the remote-ID option to the RELAY-FORWARD packet occurs automatically and no user configuration is necessary.

The DHCPv6 server does not need to echo the remote-ID option in the RELAY-REPLY packet. The Internet Assigned Numbers Authority (IANA) has assigned the DHCPv6 option code 37 for the relay agent remote-ID option.

If the remote-ID option is included in the RELAY-REPLY packet, the option is removed from the packet before it is relayed to the client.

DHCPv6 Relay Options: Reload Persistent Interface ID

The DHCPv6 Relay—Reload Persistent Interface ID Option feature makes the interface ID option persistent. The interface ID is used by relay agents to decide which interface should be used to forward a RELAY-REPLY packet. A persistent interface-ID option will not change if the device acting as a relay agent goes offline during a reload or a power outage. When the device acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as, when the relay agent reboots and the number of interfaces in the interface index changes, or when the relay agents boot up and has more virtual interfaces than it did before the reboot). This feature prevents such scenarios from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as the short form of the interface name. The interface name as the DHCPv6 interface ID helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

DHCPv6 Relay Chaining

DHCPv6 messages can be relayed through multiple relay agents. This configuration is called *relay chaining*. A relay chaining configuration can be supported only when each relay agent adds information to DHCPv6 messages before relaying them. The information helps in relaying the DHCPv6 reply back to the DHCPv6 client through the same path.

The delegated IPv6 prefix must be routable in order to be useful. The actual DHCPv6 Prefix Delegation (PD) client may not be permitted to inject routes into the delegating network. In service provider (SP) networks, for example, an edge device typically acts as a DHCPv6 relay agent, and this edge device often has the

responsibility to maintain routes within the SP network for clients' PD bindings. In the event that DHCPv6 requests and responses are relayed through a chain of DHCPv6 relays, there may be a need to introduce appropriate routes (particularly with DHCPv6 PD) in the Forwarding Information Base (FIB) so that routing is handled transparently.

DHCPv6 Relay Agent Notification for Prefix Delegation

The DHCPv6 relay agent notification for prefix delegation allows the device working as a DHCPv6 relay agent to find prefix delegation options by reviewing the contents of a DHCPv6 RELAY-REPLY packet that is relayed by the relay agent to the client. When a prefix delegation option is found by the relay agent, the relay agent extracts the information about the prefix that is being delegated and inserts an IPv6 static route matching the prefix delegation information onto the relay agent. Future packets destined to that prefix via relay will be forwarded based on the information contained in the prefix delegation. The IPv6 static route is then left in the routing table until the prefix delegation lease time expires or the relay agent receives a release packet from the client releasing the prefix delegation.

No user configuration is required for this feature. Static route management is done automatically by the relay agent.

IPv6 routes are added when the relay agent relays a RELAY-REPLY packet, and IPv6 routes are deleted when the prefix delegation lease time expires or the relay agent receives a release message. An IPv6 static route in the routing table of the relay agent can be updated when the prefix delegation lease time is extended.

The DHCP—DHCPv6 Relay Agent Notification for Prefix Delegation feature leaves a static IPv6 route on the routing table of the relay agent. The registered IPv6 address allows unicast reverse packet forwarding (uRPF) to work by allowing the device doing the reverse lookup to confirm that the IPv6 address on the relay agent is not malformed or spoofed. The static route that remains in the routing table of the relay agent can be redistributed to other routing protocols to advertise the subnets to other nodes. Static routes will be removed when a DHCP_DECLINE message is sent by the client.

DHCPv6 Relay Options: Remote ID for Ethernet Interfaces

This feature adds the remote identification (remote-ID) option to relayed (RELAY-FORWARD) DHCPv6 packets.

The remote-ID option provides information to the DHCPv6 server, including port information, the system's DUID, and the VLAN ID. Collectively, this information can be used to uniquely identify both the relay and the port on the relay through which the client's packet arrived. The DHCPv6 server uses this information to select parameters specific to a particular user, host, or subscriber modem. This feature works only for Ethernet interfaces at this time.

This feature introduces no user configuration. Because the addition of the remote-ID option to the RELAY-FORWARD packet occurs automatically, no user configuration is necessary.

The DHCPv6 server does not need to echo the remote-ID option in the RELAY-REPLY packet. Internet Assigned Numbers Authority (IANA) has assigned the DHCPv6 option code 37 for the relay agent remote-ID option.

If the remote-ID option is included in the RELAY-REPLY packet, the option is stripped out of the packet before the packet is relayed to the client.

DHCPv6 Relay Options: Reload Persistent Interface ID Option

This feature makes the interface-ID option, which is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet, persistent. A persistent interface-ID option will not change

if the router acting as a relay agent goes offline (such as during a reload or a power outage). When the router acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as cases where the relay agent reboots and has a change in the number of interfaces in the interface index, or the relay agents boots up and has more virtual interfaces than it did before the reboot). This feature prevents this scenario from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as simply the short form of the interface name. This syntax helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

DHCPv6 Relay Chaining

This feature makes the interface-ID option, which is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet, persistent. A persistent interface-ID option will not change if the router acting as a relay agent goes offline (such as during a reload or a power outage). When the router acting as a relay agent returns online, it is possible that changes to the internal interface index of the relay agent may have occurred in certain scenarios (such as cases where the relay agent reboots and has a change in the number of interfaces in the interface index, or the relay agents boots up and has more virtual interfaces than it did before the reboot). This feature prevents this scenario from causing any problems.

This feature changes the DHCPv6 interface-ID option to be expressed as simply the short form of the interface name. This syntax helps avoid potential problems that could arise due to physical or logical interfaces changing on the relay agent after a reload.

How to Configure IPv6 Access Services: DHCPv6 Relay Agent

Configuring the DHCPv6 Relay Agent

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 enable**
5. **ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 4/2/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 5	**ipv6 dhcp relay destination** *ipv6-address* [*interface-type interface-number*] **Example:** `Device(config-if) ipv6 dhcp relay destination FE80::250:A2FF:FEBF:A056 gigabitethernet 4/3/0`	Specifies a destination address to which client packets are forwarded and enables the DHCPv6 relay service on the interface.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for IPv6 Access Services: DHCPv6 Relay Agent

Example: Configuring the DHCPv6 Relay Agent

```
Device# show ipv6 dhcp interface

Ethernet1/0 is in relay mode
  Relay destinations:
    3FFB:C00:C18:6:A8BB:CCFF:FE03:2701
Serial3/0 is in relay mode
  Relay destinations:
    3FFB:C00:C18:6:A8BB:CCFF:FE03:2600
    FE80::A8BB:CCFF:FE03:2801 on Serial3/0
    FF05::1:3
```

Additional References

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

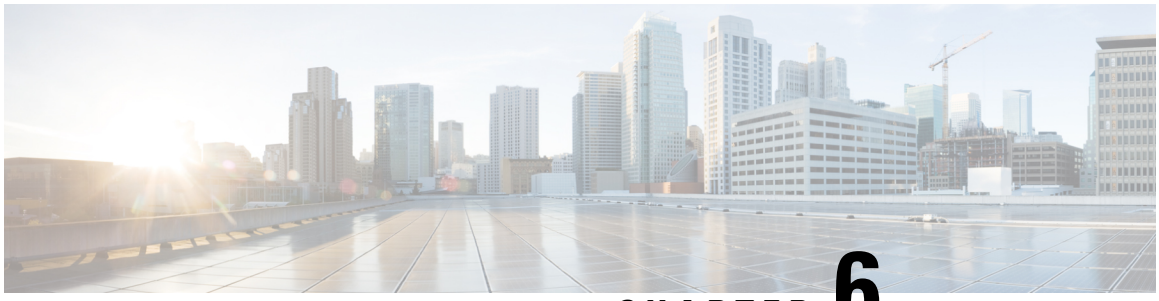
Feature Information for IPv6 Access Services: DHCPv6 Relay Agent

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7: Feature Information for IPv6 Access Services: DHCPv6 Relay Agent

Feature Name	Releases	Feature Configuration Information
IPv6 Access Services: DHCPv6 Relay Agent	Cisco IOS XE Release 3.14.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D, ASR-920-10SZ-PD, ASR-920-24SZ-IM, ASR-920-24SZ-M, ASR-920-24TZ-M).



CHAPTER 6

IPv6 Access Services: DHCPv6 Prefix Delegation

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) prefix delegation feature can be used to manage link, subnet, and site addressing changes.

Restrictions for IPv6 Access Services: DHCPv6 Prefix Delegation

- DHCPv6 Reconfigure message type is not be supported on the Cisco ASR920 Series routers.
- The **service dhcp** and **no service dhcp** commands are not applicable for DHCPv6.
- The **clear ipv6 dhcp statistics** command is not supported on the Cisco ASR920 Series routers.
- In case of IPv6, details of the default gateway are received from router advertisements (RA) and not DHCP.
- The ipv6 dhcp server automatic should be used only with IANA and not IA_PD.

Information About IPv6 Access Services: DHCPv6 Prefix Delegation

DHCPv6 Prefix Delegation

The IPv6 Access Services—DHCPv6 Prefix Delegation feature can be used to manage link, subnet, and site addressing changes. Dynamic Host Configuration Protocol for IPv6 (DHCPv6) can be used in environments to deliver stateful and stateless information, which are defined as follows:

- Stateful prefix delegation—Address assignment is centrally managed and clients must obtain configuration information such as address autoconfiguration and neighbor discovery that is not available through protocols.
- Stateless prefix delegation—Stateless configuration parameters do not require a server to maintain any dynamic state for individual clients, such as Domain Name System (DNS) server addresses and domain search list options.

Extensions to DHCPv6 also enable prefix delegation, through which an ISP can automate the process of assigning prefixes to a customer for use within the customer's network. The prefix delegation occurs between a provider edge (PE) device and customer premises equipment (CPE) using the DHCPv6 prefix delegation option. Once the ISP has delegated prefixes to a customer, the customer may further subnet and assign prefixes to the links in the customer's network.

Configuring Nodes Without Prefix Delegation

Stateless DHCPv6 allows DHCPv6 to be used for configuring a node with parameters that do not require a server to maintain any dynamic state for the node. The use of stateless DHCP is controlled by router advertisement (RA) messages multicasted by routers. The Cisco IOS XE DHCPv6 client will invoke stateless DHCPv6 when it receives an RA. The Cisco IOS XE DHCPv6 server will respond to a stateless DHCPv6 request with configuration parameters, such as the DNS servers and domain search list options.

Client and Server Identification

Each DHCPv6 client and server is identified by a DHCP unique identifier (DUID). The DUID is carried in client identifier and server identifier options. The DUID is unique across all DHCP clients and servers, and it is stable for any specific client or server. DHCPv6 uses DUIDs based on link-layer addresses for both the client and server identifier. The device uses the MAC address from the lowest-numbered interface to form the DUID. The network interface is assumed to be permanently attached to the device.

When a DHCPv6 client requests two prefixes with the same DUID but with different identity association identifiers (IAIDs) on two different interfaces, these prefixes are considered to be for two different clients, and the interface information is maintained for both.

Rapid Commit

The DHCPv6 client can obtain configuration parameters from a server either through a rapid two-message exchange (solicit, reply) or through a four-message exchange (solicit, advertise, request, and reply). By default, the four-message exchange is used. When the rapid-commit option is enabled by both the client and the server, the two-message exchange is used.

DHCPv6 Client, Server, and Relay Functions

The DHCPv6 client, server, and relay functions are mutually exclusive on an interface. When one of these functions is enabled and a user tries to configure a different function on the same interface, one of the following messages is displayed: "Interface is in DHCP client mode," "Interface is in DHCP server mode," or "Interface is in DHCP relay mode."

The following sections describe these functions:

Client Function

The DHCPv6 client function can be enabled on individual IPv6-enabled interfaces.

Note You need APPX license package to enable the DHCPv6 client function on the device.

The DHCPv6 client can request and accept those configuration parameters that do not require a server to maintain any dynamic state for individual clients, such as DNS server addresses and domain search list options.

The DHCPv6 client can also request the delegation of prefixes. The prefixes acquired from a delegating device will be stored in a local IPv6 general prefix pool. The prefixes in the general prefix pool can then be referred to from other applications; for example, the general prefix pool can be used to number device downstream interfaces.

For IPv6, the details of the default gateway are received from router advertisement (RA) and not from the DHCP server. The details of the default gateway obtained from the DHCP server are not added in the client server. Hence, the following IPv6 commands must be configured on the client interface:

- **pv6 nd autoconfig prefix"**
- **"ipv6 nd autoconfig default"**

Server Selection

A DHCPv6 client builds a list of potential servers by sending a solicit message and by collecting advertise message replies from servers. These messages are ranked based on the preference value, and servers may add a preference option to their advertise messages explicitly stating their preference value. If the client needs to acquire prefixes from servers, only servers that have advertised prefixes are considered.

IAPD and IAID

An Identity Association for Prefix Delegation (IAPD) is a collection of prefixes assigned to a requesting device. A requesting device may have more than one IAPD; for example, one for each of its interfaces.

Each IAPD is identified by an IAID. The IAID is chosen by the requesting device and is unique among the IAPD IAIDs on the requesting device. IAIDs are made consistent across reboots by using information from the associated network interface, which is assumed to be permanently attached to the device.

Server Function

The DHCPv6 server function can be enabled on individual IPv6-enabled interfaces.

The DHCPv6 server can provide configuration parameters that do not require the server to maintain any dynamic state for individual clients, such as DNS server addresses and domain search list options. The DHCPv6 server may be configured to perform prefix delegation.

All the configuration parameters for clients are independently configured into DHCPv6 configuration pools, which are stored in the NVRAM. A configuration pool can be associated with a particular DHCPv6 server on an interface when it is started. Prefixes that are to be delegated to clients may be specified either as a list of preassigned prefixes for a particular client or as IPv6 local prefix pools that are also stored in the NVRAM. The list of manually configured prefixes or IPv6 local prefix pools can be referenced and used by DHCPv6 configuration pools.

The DHCPv6 server maintains an automatic binding table in memory to track the assignment of some configuration parameters, such as prefixes between the server and its clients. Automatic bindings can be stored permanently in the database agent, such as a remote TFTP server or a local NVRAM file system.

Configuration Information Pool

A DHCPv6 configuration information pool is a named entity that includes information about available configuration parameters and policies that the control assignment of the parameters to clients from the pool. A pool is configured independently and is associated with the DHCPv6 service through the CLI.

Each configuration pool can contain the following configuration parameters and operational information:

- Prefix delegation information, which includes:
  - A prefix pool name and associated preferred and valid lifetimes
  - A list of available prefixes for a particular client and associated preferred and valid lifetimes
- A list of IPv6 addresses of DNS servers
- A domain search list, which is a string containing domain names for the DNS resolution

DHCP for IPv6 Address Assignment

DHCPv6 enables DHCP servers to pass configuration parameters, such as IPv6 network addresses, to IPv6 clients. The DHCPv6 Individual Address Assignment feature manages nonduplicate address assignment in the correct prefix based on the network where the host is connected. Assigned addresses can be from one or multiple prefix pools. Additional options, such as the default domain and DNS name-server address, can be passed back to the client. Address pools can be assigned for use on a specific interface or on multiple interfaces, or the server can automatically find the appropriate pool.

Prefix Assignment

A prefix-delegating router (DHCPv6 server) selects prefixes to be assigned to a requesting router (DHCPv6 client) upon receiving a request from the client. The server can select prefixes for a requesting client by using static and dynamic assignment mechanisms. Administrators can manually configure a list of prefixes and associated preferred and valid lifetimes for an IAPD of a specific client that is identified by its DUID.

When the delegating router receives a request from a client, it checks if there is a static binding configured for the IAPD in the client's message. If a static binding is present, the prefixes in the binding are returned to the client. If no such binding is found, the server attempts to assign prefixes for the client from other sources.

The Cisco IOS XE DHCPv6 server can assign prefixes dynamically from an IPv6 local prefix pool. When the server receives a prefix request from a client, it attempts to obtain unassigned prefixes from the pool. After the client releases the previously assigned prefixes, the server returns them to the pool for reassignment.

An IPv6 prefix delegating router can also select prefixes for a requesting router based on an external authority such as a RADIUS server using the Framed-IPv6-Prefix attribute.

Automatic Binding

Each DHCPv6 configuration pool has an associated binding table. The binding table contains records of all prefixes in the configuration pool that have been explicitly delegated to clients. Each entry in the binding table contains the following information:

- Client DUID.
- Client IPv6 address.
- A list of IAPDs associated with the client.
- A list of prefixes delegated to each IAPD.

- Preferred and valid lifetimes for each prefix.
- The configuration pool to which this binding table belongs.
- The network interface on which the server that is using the pool is running.

A binding table entry is automatically created whenever a prefix is delegated to a client from the configuration pool, and the entry is updated when the client renews, rebinds, or confirms the prefix delegation. A binding table entry is deleted when the client voluntarily releases all the prefixes in the binding, the valid lifetimes of all prefixes have expired, or administrators run the **clear ipv6 dhcp binding** command.

Binding Database

Each permanent storage to which the binding database is saved is called the database agent. A database agent can be a remote host, such as an FTP server, or a local file system, such as the NVRAM.

Automatic bindings are maintained in the RAM and can be saved to some permanent storage so that information about configurations, such as prefixes assigned to clients, is not lost after a system reload. The bindings are stored as text records for easy maintenance. Each record contains the following information:

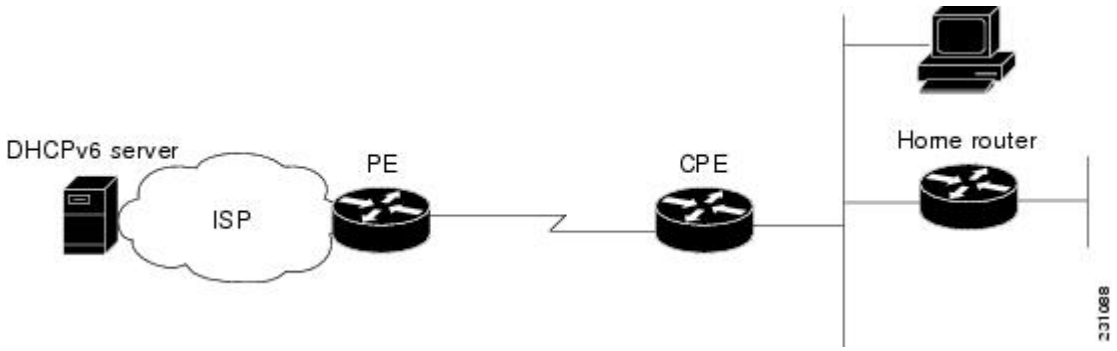
- DHCPv6 pool name from which the configuration was assigned to the client.
- Interface identifier from which the client requests were received.
- The client IPv6 address.
- The client DUID.
- IAID of the IAPD.
- Prefix delegated to the client.
- The prefix length.
- The prefix preferred lifetime in seconds.
- The prefix valid lifetime in seconds.
- The prefix expiration time stamp.
- Optional local prefix pool name from which the prefix was assigned.

DHCPv6 Server Stateless Autoconfiguration

Hierarchical DHCPv6 for stateless configuration parameters allows a stateless or stateful DHCPv6 client to export configuration parameters (DHCPv6 options) to a local DHCPv6 server pool. The local DHCPv6 server can then provide the imported configuration parameters to other DHCPv6 clients.

The figure below shows a typical broadband deployment.

Figure 6: Broadband Topology



The CPE interface towards the PE can be a stateless or stateful DHCPv6 client. In either case, the ISP-side DHCPv6 server may provide configuration parameters such as DNS server addresses, domain names, and Simple Network Time Protocol (SNTP) servers to the DHCP client on the CPE. Such information can be specific to ISPs.

In addition to being a DHCPv6 client (for example, towards the ISP), the CPE may act as a DHCPv6 server to the home network. For example, neighbor discovery followed by a stateless or stateful DHCPv6 client can occur on the link between the CPE and the home devices (such as the home router or PC). In some cases, the information to be provided to the home network is the same as that obtained from the ISP-side DHCPv6 server. Because this information can be dynamically changed, it cannot be hard-configured in the CPE's configuration. Therefore, the DHCPv6 component on the CPE allows automatic importing of configuration parameters from the DHCPv6 client to the DHCPv6 server pool.

DHCPv6 supports the following options for IPv6 on the server:

Information Refresh Server Option

The DHCPv6 information refresh option can specify a maximum limit for the length of time a client should wait before refreshing the information retrieved from DHCPv6. This option is used with stateless DHCPv6 because there are no addresses or other entities with lifetimes that can tell the client when to contact the DHCPv6 server to refresh its configuration.

NIS- and NIS+-Related Server Options

Users can configure the network information service (NIS) or NIS plus (NIS+) address or domain name of a DHCPv6 server using NIS- and NIS+-related options, and then import that information to the DHCPv6 client.

SIP Server Options

Session Initiation Protocol (SIP) server options contain either a list of domain names or a list of IPv6 addresses that can be mapped to one or more SIP outbound proxy servers. One option carries a list of domain names, and the other option carries a list of 128-bit IPv6 addresses.

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions or calls. A SIP system has several logical components: user agents, proxy servers, redirect servers, and registrars. User agents may contain SIP clients; proxy servers always contain SIP clients.

SNTP Server Option

The SNTP server option provides a list of one or more IPv6 addresses of SNTP servers available to the client for synchronization. Clients use these SNTP servers to synchronize their system time to that of the standard

time servers. The DHCPv6 server may list the SNTP servers in decreasing order of preference, but clients treat the list of SNTP servers as an ordered list.

Relay Function

A DHCPv6 relay agent, which may reside on the link of the DHCP client, is used to relay messages between the client and the server. The DHCPv6 relay agent operation is transparent to the client. A DHCPv6 client locates a DHCPv6 server using a reserved, link-scoped multicast address. For direct communication between the DHCPv6 client and the DHCPv6 server, both of them must be attached to the same link. However, in some situations where ease of management, economy, or scalability is a concern, it is desirable to allow a DHCPv6 client to send a message to a DHCPv6 server that is not connected to the same line.

DHCPv6 messages can be relayed through multiple relay agents. This configuration is called relay chaining. A relay-chaining configuration can be supported only when each relay agent adds information to the DHCPv6 messages before relaying them. This information helps in relaying the DHCPv6 reply back to the DHCPv6 client through the same path.

For more information on *Configuring the DHCPv6 Relay Agent*, see IPV6 Access Services: DHCPv6 Relay Agent chapter.

How to Configure IPv6 Access Services: DHCPv6 Prefix Delegation

Configuring the DHCPv6 Server Function

Configuring the DHCPv6 Configuration Pool

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **domain-name** *domain*
5. **dns-server** *ipv6-address*
6. **prefix-delegation** *ipv6-prefix / prefix-length client-duid* [**iaid** *iaid*] [*lifetime*]
7. **prefix-delegation pool** *poolname* [**lifetime** *valid-lifetime preferred-lifetime*]
8. **exit**
9. **interface** *type number*
10. **ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp pool** *poolname* **Example:** `Device(config)# ipv6 dhcp pool pool1`	Configures a DHCPv6 configuration information pool and enters DHCPv6 pool configuration mode. The *pool name* can be a string, such as "abcd" or an integer value, such as 0. During execution, the configuration mode changes to DHCPv6 pool configuration mode. In this mode, you can configure pool parameters, such as prefixes to be delegated, DNS servers, and so on.
Step 4	**domain-name** *domain* **Example:** `Device(config-dhcp)# domain-name example.com`	Configures a domain name for a DHCPv6 client.
Step 5	**dns-server** *ipv6-address* **Example:** `Device(config-dhcp)# dns-server 2001:DB8:3000:3000::42`	Specifies the DNS IPv6 servers available to a DHCPv6 client.
Step 6	**prefix-delegation** *ipv6-prefix* / *prefix-length client-duid* [**iaid** *iaid*] [*lifetime*] **Example:** `Device(config-dhcp)# prefix-delegation 2001:DB8:1263::/48 0005000400F1A4D070D03`	Specifies a manually configured numeric prefix to be delegated to a specified client's IAPD.
Step 7	**prefix-delegation pool** *poolname* [**lifetime** *valid-lifetime preferred-lifetime*] **Example:** `Device(config-dhcp)# prefix-delegation pool pool1 lifetime 1800 60`	Specifies a named IPv6 local prefix pool from which prefixes are delegated to DHCPv6 clients. The lifetime values are *valid-lifetime* and *preferred-lifetime*. These are referred to as T1 and T2. When the T2 expires, a renew request is sent to the particular server and if the client does not get a response within T1, the client sends a REBIND request to all available servers. The value of **lifetime** can be specified as: • a fixed duration that remains constant across consecutive advertisements

	Command or Action	Purpose
		• absolute expiration time in the future, so that the advertised lifetime decrements in real time and is equal to zero at the specified time. The specified duration is between 60 and 4294967295 seconds or infinity if the keyword **infinite** is specified. If the lifetimes are not specified, by default, the *valid-lifetime* is 2592000 seconds (or 30 days) and the *preferred-lifetime* is 604800 seconds (or 7 days).
Step 8	**exit** **Example:** `Device(config-dhcp)# exit`	Exits DHCPv6 pool configuration mode, and returns the device to global configuration mode.
Step 9	**interface** *type number* **Example:** `Device(config)# interface serial 3`	Specifies an interface type and number, and enters interface configuration mode.
Step 10	**ipv6 dhcp server** *poolname* [**rapid-commit**] [**preference** *value*] [**allow-hint**] **Example:** `Device(config-if)# ipv6 dhcp server pool1`	Enables or disables DHCPv6 service on an interface. • If specified, **rapid-commit** enables the use of the two-message exchange for prefix delegation and other configuration. If the **rapid-commit** option is used in the Solicit message and **rapid-commit** is enabled for the server, the server responds to the Solicit with a Reply message. By default, **rapid-commit** is disabled. • Default value of **preference** is 0. • If the **allow-hint** option is specified, is a valid prefix in the associated local prefix pool and is not assigned to anybody, the server delegates the client-suggested prefix in the Solicit and Request messages. Otherwise, the hint is ignored and a prefix is delegated from the free list in the pool. **Note** By default, DHCPv6 service on an interface is disabled.
Step 11	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring a Binding Database Agent for the Server Function

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp database** *agent* [**write-delay** *seconds*] [**timeout** *seconds*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 dhcp database** *agent* [**write-delay** *seconds*] [**timeout** *seconds*] **Example:** `Device(config)# ipv6 dhcp database tftp://10.0.0.1/dhcp-binding`	Specifies DHCPv6 binding database agent parameters. • • **agent-URL**—flash, NVRAM, FTP, TFTP, or RCP uniform resource locator. • **write-delay**—specifies how often DHCP sends database updates. By default, DHCPv6 server waits 300 seconds before transmitting database changes. The minimum delay is 60 seconds. • **timeout**—specifies the time to wait for a database transfer. Infinity is defined as zero seconds. Transfers that exceed the timeout period are aborted. Default value is 300 seconds.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring the DHCPv6 Client Function

General prefixes can be defined dynamically from a prefix received by a DHCPv6 prefix delegation client. The delegated prefix is stored in a general prefix.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 dhcp client pd** {*prefix-name* | **hint** *ipv6-prefix*} [**rapid-commit**]
5. **ipv6 nd autoconfig prefix**
6. **ipv6 nd autoconfig default-router**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ipv6 dhcp client pd** {*prefix-name* | **hint** *ipv6-prefix*} [**rapid-commit**] **Example:** `Device(config-if)# ipv6 dhcp client pd dhcp-prefix`	Enables the DHCPv6 client process and enables a request for prefix delegation through a specified interface. **Note** The **ipv6 dhcp client pd hint** command should always be used along with the **ipv6 dhcp client pd** *pd-name* command.
Step 5	**ipv6 nd autoconfig prefix** **Example:** `Device(config-if)# ipv6 nd autoconfig prefix`	Allows Neighbor Discovery to install all valid on-link prefixes from router advertisements (RAs) received on the interface.
Step 6	**ipv6 nd autoconfig default-router** **Example:** `Device(config-if)# ipv6 nd autoconfig default-router`	Allows Neighbor Discovery to install a default route to the Neighbor Discovery-derived default router.
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Example

The following example depicts the usage of **ipv6 dhcp client pd hint** command along with the **ipv6 dhcp client pd** *pd-name* command.

```
interface GigabitEthernet0/10
no ip address
media-type auto-select
negotiation auto
ipv6 address prefix-from-provider ::1/48
ipv6 enable
ipv6 dhcp client pd hint 2001:DB8:43::/48
ipv6 dhcp client pd prefix-from-provider
```

Deleting Automatic Client Bindings from the DHCPv6 Binding Table

SUMMARY STEPS

1. **enable**
2. **clear ipv6 dhcp binding** [*ipv6-address*] [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 dhcp binding** [*ipv6-address*] [**vrf** *vrf-name*] **Example:** `Device# clear ipv6 dhcp binding`	Deletes automatic client bindings from the DHCPv6 binding table.

Removing Previously-Acquired Prefixes

SUMMARY STEPS

1. **enable**
2. **clear ipv6 dhcp client** *interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**clear ipv6 dhcp client** *interface* **Example:** `Device# clear ipv6 dhcp client`	Restarts DHCPv6 client on an interface after releasing and un-configuring previously-acquired prefixes and other configuration options.

Debugging DHCPv6 Binding Database

SUMMARY STEPS

1. **enable**
2. **debug ipv6 dhcp database**
3. **debug ipv6 dhcp detail**
4. **debug ipv6 dhcp relay**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ipv6 dhcp database** **Example:** `Device# debug ipv6 dhcp database`	Enables or disables debugging of DHCPv6 binding database.
Step 3	**debug ipv6 dhcp detail** **Example:** `Device# debug ipv6 dhcp detail`	Displays the client and server packet details.
Step 4	**debug ipv6 dhcp relay** **Example:** `Device# debug ipv6 dhcp relay`	Displays the relay details.

Configuration Examples for IPv6 Access Services: DHCPv6 Prefix Delegation

Example: Configuring the DHCPv6 Server Function

DHCPv6 clients are connected to the DHCPv6 server on Gigabit Ethernet interface 0/0/0. The server is configured to use parameters from the DHCP pool called dhcp-pool. This pool provides clients with the IPv6 address of a DNS server and the domain name to be used. It also specifies that prefixes can be delegated from the prefix pool called client-prefix-pool1. The prefixes delegated will have valid and preferred lifetimes of 1800 and 600 seconds respectively. The prefix pool named client-prefix-pool1 has a prefix of length /40 from which it will delegate (sub) prefixes of length /48.

```
ipv6 dhcp pool dhcp-pool
 prefix-delegation pool client-prefix-pool1 lifetime 1800 600
 dns-server 2001:DB8:3000:3000::42
 domain-name example.com
!
interface GigabitEthernet0/0
 description downlink to clients
 ipv6 address FEC0:240:104:2001::139/64
 ipv6 dhcp server dhcp-pool
!
ipv6 local pool client-prefix-pool1 2001:DB8:1200::/40 48
```

The following example from the **show ipv6 dhcp** command shows the DUID of the device:

```
Router# show ipv6 dhcp

This device's DHCPv6 unique identifier(DUID): 000300010002FCA5DC1C
```

In the following example, the show ipv6 dhcp binding command shows information about two clients, including their DUIDs, IAPDs, prefixes, and preferred and valid lifetimes:

```
Router# show ipv6 dhcp binding

Client: FE80::202:FCFF:FEA5:DC39 (GigabitEthernet2/1/0)
  DUID: 000300010002FCA5DC1C
  IA PD: IA ID 0x00040001, T1 0, T2 0
    Prefix: 3FFE:C00:C18:11::/68
            preferred lifetime 180, valid lifetime 12345
            expires at Nov 08 2002 02:24 PM (12320 seconds)
Client: FE80::202:FCFF:FEA5:C039 (GigabitEthernet2/1/0)
  DUID: 000300010002FCA5C01C
  IA PD: IA ID 0x00040001, T1 0, T2 0
    Prefix: 3FFE:C00:C18:1::/72
            preferred lifetime 240, valid lifetime 54321
            expires at Nov 09 2002 02:02 AM (54246 seconds)
    Prefix: 3FFE:C00:C18:2::/72
            preferred lifetime 300, valid lifetime 54333
            expires at Nov 09 2002 02:03 AM (54258 seconds)
    Prefix: 3FFE:C00:C18:3::/72
            preferred lifetime 280, valid lifetime 51111
```

In the following example, the **show ipv6 dhcp database** command provides information on the binding database agents TFTP, NVRAM, and flash:

```
Router# show ipv6 dhcp database

Database agent tftp://172.19.216.133/db.tftp:
  write delay: 69 seconds, transfer timeout: 300 seconds
  last written at Jan 09 2003 01:54 PM,
     write timer expires in 56 seconds
  last read at Jan 06 2003 05:41 PM
  successful read times 1
  failed read times 0
  successful write times 3172
  failed write times 2
Database agent nvram:/dhcpv6-binding:
  write delay: 60 seconds, transfer timeout: 300 seconds
  last written at Jan 09 2003 01:54 PM,
     write timer expires in 37 seconds
  last read at never
  successful read times 0
  failed read times 0
  successful write times 3325
  failed write times 0
Database agent flash:/dhcpv6-db:
  write delay: 82 seconds, transfer timeout: 3 seconds
  last written at Jan 09 2003 01:54 PM,
    write timer expires in 50 seconds
  last read at never
  successful read times 0
  failed read times 0
  successful write times 2220
  failed write times 614
```

Example: Configuring the DHCPv6 Configuration Pool

In the following example, the **show ipv6 dhcp pool** command provides information on the configuration pool named svr-p1, including the static bindings, prefix information, the DNS server, and the domain names found in the svr-p1 pool:

```
Router# show ipv6 dhcp pool

DHCPv6 pool: svr-p1
  Static bindings:
    Binding for client 000300010002FCA5C01C
      IA PD: IA ID 00040002,
        Prefix: 3FFE:C00:C18:3::/72
                preferred lifetime 604800, valid lifetime 2592000
      IA PD: IA ID not specified; being used by 00040001
        Prefix: 3FFE:C00:C18:1::/72
                preferred lifetime 240, valid lifetime 54321
        Prefix: 3FFE:C00:C18:2::/72
                preferred lifetime 300, valid lifetime 54333
        Prefix: 3FFE:C00:C18:3::/72
                preferred lifetime 280, valid lifetime 51111
  Prefix from pool: local-p1, Valid lifetime 12345, Preferred lifetime 180
  DNS server: 2001:DB8:1001::1
  DNS server: 2001:DB8:1001::2
  Domain name: example1.net
  Domain name: example2.net
  Domain name: example3.net
Active clients: 2
Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by bird
```

```
!
hostname Router
!
ip cef
ipv6 unicast-routing
ipv6 cef
ipv6 cef accounting prefix-length
!
!
interface GigabitEthernet0/0/0
 ip address 10.4.9.11 255.0.0.0
 media-type 10BaseT
 ipv6 address 2001:DB8:C18:1::/64 eui-64
```

Example: Configuring the DHCPv6 Client Function

This DHCPv6 client has three interfaces: Gigabit Ethernet interface 0/0/0 is the upstream link to a service provider, which has a DHCPv6 server function enabled. The Fast Ethernet interfaces 0/0/0 and 0/1/0 are links to local networks.

The upstream interface, Gigabit Ethernet interface 0/0/0, has the DHCPv6 client function enabled. Prefixes delegated by the provider are stored in the general prefix called prefix-from-provider.

The local networks, Fast Ethernet interfaces 0/0/0 and 0/1/0, both assign interface addresses based on the general prefix called prefix-from-provider. The bits on the left of the addresses come from the general prefix, and the bits on the right of the addresses are specified statically.

```
interface GigabitEthernet 0/0/0
 description uplink to provider DHCP IPv6 server
 ipv6 dhcp client pd prefix-from-provider
!
interface FastEthernet 0/0/0
 description local network 0
 ipv6 address prefix-from-provider ::5:0:0:0:100/64
!
interface FastEthernet 0/1/0
 description local network 1
 ipv6 address prefix-from-provider ::6:0:0:0:100/64
```

Example: Configuring a Database Agent for the Server Function

The DHCPv6 server is configured to store table bindings to the file named dhcp-binding on the server at address 10.0.0.1 using the TFTP protocol. The bindings are saved every 120 seconds.

```
ipv6 dhcp database tftp://10.0.0.1/dhcp-binding write-delay 120
```

The following example shows how to specify DHCP for IPv6 binding database agent parameters and store binding entries in bootflash:

```
ipv6 dhcp database bootflash
```

Example: Displaying DHCP Server and Client Information on the Interface

The following is sample output from the **show ipv6 dhcp interface** command. In the first example, the command is used on a router that has an interface acting as a DHCPv6 server. In the second example, the command is used on a router that has an interface acting as a DHCPv6 client:

```
Router1# show ipv6 dhcp interface

GigabitEthernet2/0 is in server mode
  Using pool: svr-p1
  Preference value: 20
  Rapid-Commit is disabled

Router2# show ipv6 dhcp interface

GigabitEthernet2/0 is in client mode
  State is OPEN (1)
  List of known servers:
    Address: FE80::202:FCFF:FEA1:7439, DUID 000300010002FCA17400
    Preference: 20
      IA PD: IA ID 0x00040001, T1 120, T2 192
        Prefix: 3FFE:C00:C18:1::/72
                preferred lifetime 240, valid lifetime 54321
                expires at Nov 08 2002 09:10 AM (54319 seconds)
        Prefix: 3FFE:C00:C18:2::/72
                preferred lifetime 300, valid lifetime 54333
                expires at Nov 08 2002 09:11 AM (54331 seconds)
        Prefix: 3FFE:C00:C18:3::/72
                preferred lifetime 280, valid lifetime 51111
                expires at Nov 08 2002 08:17 AM (51109 seconds)
      DNS server: 2001:DB8:1001::1
      DNS server: 2001:DB8:1001::2
      Domain name: example1.net
      Domain name: example2.net
      Domain name: example3.net
    Prefix name is cli-p1
    Rapid-Commit is enabled
```

Example: Debugging DHCPv6

The following is sample output from the **debug ipv6 dhcp detail** command. If the keyword **detail** is specified, a detailed DHCPv6 message decoding report is displayed.

Client Debug Logs:

```
Mar 19 12:47:25.830 IST: IPv6 DHCP: DHCPv6 changes state from IDLE to SOLICIT (START) on
GigabitEthernet0/1
Mar 19 12:47:25.879 IST: %SYS-5-CONFIG_I: Configured from console by console
Mar 19 12:47:28.582 IST: IPv6 DHCP: detailed packet contents
Mar 19 12:47:28.582 IST: src FE80::21F:C2FF:FEAD:9D8C
Mar 19 12:47:28.582 IST: dst FF02::1:2 (GigabitEthernet0/1)
Mar 19 12:47:28.583 IST: type SOLICIT(1), xid 1602672
Mar 19 12:47:28.583 IST: option ELAPSED-TIME(8), len 2
Mar 19 12:47:28.583 IST: elapsed-time 0
Mar 19 12:47:28.583 IST: option CLIENTID(1), len 10
Mar 19 12:47:28.583 IST: 00030001001FC2AD9D80
Mar 19 12:47:28.583 IST: option ORO(6), len 6
```

```
Mar 19 12:47:28.583 IST: IA-PD,DNS-SERVERS,DOMAIN-LIST
Mar 19 12:47:28.583 IST: option IA-PD(25), len 12
Mar 19 12:47:28.583 IST: IAID 0x00120001, T1 0, T2 0
Mar 19 12:47:28.583 IST: IPv6 DHCP: Sending SOLICIT to FF02::1:2 src FE80::21F:C2FF:FEAD:9D8C
 on GigabitEthernet0/1
Mar 19 12:47:29.720 IST: IPv6 DHCP: detailed packet contents
Mar 19 12:47:29.720 IST: src FE80::21F:C2FF:FEAD:9D8C
Mar 19 12:47:29.720 IST: dst FF02::1:2 (GigabitEthernet0/1)
Mar 19 12:47:29.720 IST: type SOLICIT(1), xid 1602672
Mar 19 12:47:29.720 IST: option ELAPSED-TIME(8), len 2
Mar 19 12:47:29.720 IST: elapsed-time 113
Mar 19 12:47:29.720 IST: option CLIENTID(1), len 10
Mar 19 12:47:29.720 IST: 00030001001FC2AD9D80
Mar 19 12:47:29.720 IST: option ORO(6), len 6
Mar 19 12:47:29.720 IST: IA-PD,DNS-SERVERS,DOMAIN-LIST
Mar 19 12:47:29.720 IST: option IA-PD(25), len 12
Mar 19 12:47:29.720 IST: IAID 0x00120001, T1 0, T2 0
Mar 19 12:47:29.720 IST: IPv6 DHCP: Sending SOLICIT to FF02::1:2 src FE80::21F:C2FF:FEAD:9D8C
 on GigabitEthernet0/1
Mar 19 12:47:29.746 IST: IPv6 DHCP: Received ADVERTISE message
Mar 19 12:47:29.746 IST: IPv6 DHCP: Received ADVERTISE from FE80::1EE8:5DFF:FEC5:3883 on
GigabitEthernet0/1
Mar 19 12:47:29.746 IST: IPv6 DHCP: detailed packet contents
Mar 19 12:47:29.746 IST: src FE80::1EE8:5DFF:FEC5:3883 (GigabitEthernet0/1)
Mar 19 12:47:29.746 IST: dst FE80::21F:C2FF:FEAD:9D8C (GigabitEthernet0/1)
Mar 19 12:47:29.746 IST: type ADVERTISE(2), xid 1602672
Mar 19 12:47:29.746 IST: option SERVERID(2), len 10
Mar 19 12:47:29.746 IST: 00030001F41FC2ADA080
Mar 19 12:47:29.746 IST: option CLIENTID(1), len 10
Mar 19 12:47:29.746 IST: 00030001001FC2AD9D80
Mar 19 12:47:29.746 IST: option IA-PD(25), len 41
Mar 19 12:47:29.746 IST: IAID 0x00120001, T1 300, T2 480
Mar 19 12:47:29.746 IST: option IAPREFIX(26), len 25
Mar 19 12:47:29.746 IST: preferred 600, valid 1800, prefix 2001:DB8:43::/48
Mar 19 12:47:29.746 IST: option DNS-SERVERS(23), len 16
Mar 19 12:47:29.747 IST: 2001:DB8::58
Mar 19 12:47:29.747 IST: option DOMAIN-LIST(24), len 10
Mar 19 12:47:29.747 IST: nour.com
Mar 19 12:47:29.747 IST: IPv6 DHCP: Adding server FE80::1EE8:5DFF:FEC5:3883
Mar 19 12:47:29.747 IST: IPv6 DHCP: detailed packet contents
Mar 19 12:47:29.747 IST: src FE80::21F:C2FF:FEAD:9D8C
Mar 19 12:47:29.747 IST: dst FF02::1:2 (GigabitEthernet0/1)
Mar 19 12:47:29.747 IST: type REQUEST(3), xid 1606592
Mar 19 12:47:29.747 IST: option ELAPSED-TIME(8), len 2
Mar 19 12:47:29.747 IST: elapsed-time 0
Mar 19 12:47:29.747 IST: option CLIENTID(1), len 10
Mar 19 12:47:29.747 IST: 00030001001FC2AD9D80
Mar 19 12:47:29.747 IST: option ORO(6), len 6
Mar 19 12:47:29.747 IST: IA-PD,DNS-SERVERS,DOMAIN-LIST
Mar 19 12:47:29.747 IST: option SERVERID(2), len 10
Mar 19 12:47:29.747 IST: 00030001F41FC2ADA080
Mar 19 12:47:29.747 IST: option IA-PD(25), len 41
Mar 19 12:47:29.747 IST: IAID 0x00120001, T1 0, T2 0
Mar 19 12:47:29.747 IST: option IAPREFIX(26), len 25
Mar 19 12:47:29.747 IST: preferred 0, valid 0, prefix 2001:DB8:43::/48
Mar 19 12:47:29.747 IST: IPv6 DHCP: Sending REQUEST to FF02::1:2 src FE80::21F:C2FF:FEAD:9D8C
 on GigabitEthernet0/1
Mar 19 12:47:29.747 IST: IPv6 DHCP: DHCPv6 changes state from SOLICIT to REQUEST
(ADVERTISE_RECEIVED) on GigabitEthernet0/1
Mar 19 12:47:29.754 IST: IPv6 DHCP: Received REPLY message
Mar 19 12:47:29.754 IST: IPv6 DHCP: Received REPLY from FE80::1EE8:5DFF:FEC5:3883 on
GigabitEthernet0/1
Mar 19 12:47:29.754 IST: IPv6 DHCP: detailed packet contents
Mar 19 12:47:29.754 IST: src FE80::1EE8:5DFF:FEC5:3883 (GigabitEthernet0/1)
```

```
Mar 19 12:47:29.754 IST: dst FE80::21F:C2FF:FEAD:9D8C (GigabitEthernet0/1)
Mar 19 12:47:29.754 IST: type REPLY(7), xid 1606592
Mar 19 12:47:29.754 IST: option SERVERID(2), len 10
Mar 19 12:47:29.754 IST: 00030001F41FC2ADA080
Mar 19 12:47:29.754 IST: option CLIENTID(1), len 10
Mar 19 12:47:29.754 IST: 00030001001FC2AD9D80
Mar 19 12:47:29.754 IST: option IA-PD(25), len 41
Mar 19 12:47:29.754 IST: IAID 0x00120001, T1 300, T2 480
Mar 19 12:47:29.754 IST: option IAPREFIX(26), len 25
Mar 19 12:47:29.754 IST: preferred 600, valid 1800, prefix 2001:DB8:43::/48
Mar 19 12:47:29.754 IST: option DNS-SERVERS(23), len 16
Mar 19 12:47:29.754 IST: 2001:DB8::58
Mar 19 12:47:29.754 IST: option DOMAIN-LIST(24), len 10
Mar 19 12:47:29.754 IST: nour.com
Mar 19 12:47:29.754 IST: IPv6 DHCP: Processing options
Mar 19 12:47:29.754 IST: IPv6 DHCP: Adding prefix 2001:DB8:43::/48 to prefix-from-provider
Mar 19 12:47:29.755 IST: IPv6 DHCP: T1 set to expire in 300 seconds
Mar 19 12:47:29.755 IST: IPv6 DHCP: T2 set to expire in 480 seconds
Mar 19 12:47:29.755 IST: IPv6 DHCP: Configuring DNS server 2001:DB8::58
Mar 19 12:47:29.755 IST: IPv6 DHCP: Configuring domain name nour.com
Mar 19 12:47:29.755 IST: IPv6 DHCP: DHCPv6 changes state from REQUEST to OPEN (REPLY_RECEIVED)
 on GigabitEthernet0/1
```

Server Debug Logs:

```
IPv6 DHCP: Add IAPD routes, pool LPDCONF, idb GigabitEthernet0/4/4
Mar 19 12:47:25.744 IST: IPv6 DHCP: Add IANA routes, pool LPDCONF, idb GigabitEthernet0/4
Mar 19 12:47:29.733 IST: IPv6 DHCP: Received RELAY-FORWARD from FE80::1EE8:5DFF:FEC5:3884
on GigabitEthernet0/4
Mar 19 12:47:29.733 IST: IPv6 DHCP: detailed packet contents
Mar 19 12:47:29.733 IST: src FE80::1EE8:5DFF:FEC5:3884 (GigabitEthernet0/4)
Mar 19 12:47:29.733 IST: dst 2001:DB8:12::1 (GigabitEthernet0/4)
Mar 19 12:47:29.733 IST: type RELAY-FORWARD(12), hop 0
Mar 19 12:47:29.733 IST: link ::
Mar 19 12:47:29.733 IST: peer FE80::21F:C2FF:FEAD:9D8C
Mar 19 12:47:29.733 IST: option RELAY-MSG(9), len 50
Mar 19 12:47:29.733 IST: type SOLICIT(1), xid 1602672
Mar 19 12:47:29.733 IST: option ELAPSED-TIME(8), len 2
Mar 19 12:47:29.733 IST: elapsed-time 113
Mar 19 12:47:29.733 IST: option CLIENTID(1), len 10
Mar 19 12:47:29.733 IST: 00030001001FC2AD9D80
Mar 19 12:47:29.734 IST: option ORO(6), len 6
Mar 19 12:47:29.734 IST: IA-PD,DNS-SERVERS,DOMAIN-LIST
Mar 19 12:47:29.734 IST: option IA-PD(25), len 12
Mar 19 12:47:29.734 IST: IAID 0x00120001, T1 0, T2 0
Mar 19 12:47:29.734 IST: option CLIENT-LINKLAYER-ADDRESS(79), len 8
Mar 19 12:47:29.734 IST: 001f.c2ad.9d8c
Mar 19 12:47:29.734 IST: option INTERFACE-ID(18), len 7
Mar 19 12:47:29.734 IST: 0x4769302F302F33
Mar 19 12:47:29.734 IST: option REMOTEID(37), len 22
Mar 19 12:47:29.734 IST: 0x000000090200030000000000A000300011CE85DC53880
Mar 19 12:47:29.734 IST: IPv6 DHCP: Using interface pool LPDCONF
Mar 19 12:47:29.734 IST: IPv6 DHCP_AAA: Retrieved subblock; It has AAA DNS_SERVERS=0
Mar 19 12:47:29.734 IST: IPv6 DHCP: Option CLIENT-LINKLAYER-ADDRESS(79) in Relay-forward
ignored: 0001001FC2AD9D8C
Mar 19 12:47:29.734 IST: IPv6 DHCP: Option REMOTEID(37):
000000090200030000000000A000300011CE85DC53880
Mar 19 12:47:29.734 IST: IPv6 DHCP: SAS retured Null falling to link local
Mar 19 12:47:29.734 IST: IPv6 DHCP: Returning Link local address FE80::F61F:C2FF:FEAD:A0A8
Mar 19 12:47:29.734 IST: IPv6 DHCP: detailed packet contents
Mar 19 12:47:29.734 IST: src FE80::F61F:C2FF:FEAD:A0A8
Mar 19 12:47:29.734 IST: dst FE80::1EE8:5DFF:FEC5:3884 (GigabitEthernet0/4)
Mar 19 12:47:29.735 IST: type RELAY-REPLY(13), hop 0
Mar 19 12:47:29.735 IST: link ::
```

```
Mar 19 12:47:29.735 IST: peer FE80::21F:C2FF:FEAD:9D8C
Mar 19 12:47:29.735 IST: option RELAY-MSG(9), len 111
Mar 19 12:47:29.735 IST: type ADVERTISE(2), xid 1602672
Mar 19 12:47:29.735 IST: option SERVERID(2), len 10
Mar 19 12:47:29.735 IST: 00030001F41FC2ADA080
Mar 19 12:47:29.735 IST: option CLIENTID(1), len 10
Mar 19 12:47:29.735 IST: 00030001001FC2AD9D80
Mar 19 12:47:29.735 IST: option IA-PD(25), len 41
Mar 19 12:47:29.735 IST: IAID 0x00120001, T1 300, T2 480
Mar 19 12:47:29.735 IST: option IAPREFIX(26), len 25
Mar 19 12:47:29.735 IST: preferred 600, valid 1800, prefix 2001:DB8:43::/48
Mar 19 12:47:29.735 IST: option DNS-SERVERS(23), len 16
Mar 19 12:47:29.735 IST: 2001:DB8::58
Mar 19 12:47:29.735 IST: option DOMAIN-LIST(24), len 10
Mar 19 12:47:29.735 IST: nour.com
Mar 19 12:47:29.735 IST: option INTERFACE-ID(18), len 7
Mar 19 12:47:29.735 IST: 0x4769302F302F33
Mar 19 12:47:29.735 IST: IPv6 DHCP: Sending RELAY-REPLY to FE80::1EE8:5DFF:FEC5:3884 src
FE80::F61F:C2FF:FEAD:A0A8 on GigabitEthernet0/4
Mar 19 12:47:29.750 IST: IPv6 DHCP: Received RELAY-FORWARD from FE80::1EE8:5DFF:FEC5:3884
on GigabitEthernet0/4
Mar 19 12:47:29.750 IST: IPv6 DHCP: detailed packet contents
Mar 19 12:47:29.750 IST: src FE80::1EE8:5DFF:FEC5:3884 (GigabitEthernet0/4)
Mar 19 12:47:29.750 IST: dst 2001:DB8:12::1 (GigabitEthernet0/4)
Mar 19 12:47:29.750 IST: type RELAY-FORWARD(12), hop 0
Mar 19 12:47:29.750 IST: link ::
Mar 19 12:47:29.750 IST: peer FE80::21F:C2FF:FEAD:9D8C
Mar 19 12:47:29.750 IST: option RELAY-MSG(9), len 93
Mar 19 12:47:29.750 IST: type REQUEST(3), xid 1606592
Mar 19 12:47:29.750 IST: option ELAPSED-TIME(8), len 2
Mar 19 12:47:29.750 IST: elapsed-time 0
Mar 19 12:47:29.750 IST: option CLIENTID(1), len 10
Mar 19 12:47:29.750 IST: 00030001001FC2AD9D80
Mar 19 12:47:29.750 IST: option ORO(6), len 6
Mar 19 12:47:29.750 IST: IA-PD,DNS-SERVERS,DOMAIN-LIST
Mar 19 12:47:29.750 IST: option SERVERID(2), len 10
Mar 19 12:47:29.750 IST: 00030001F41FC2ADA080
Mar 19 12:47:29.751 IST: option IA-PD(25), len 41
Mar 19 12:47:29.751 IST: IAID 0x00120001, T1 0, T2 0
Mar 19 12:47:29.751 IST: option IAPREFIX(26), len 25
Mar 19 12:47:29.751 IST: preferred 0, valid 0, prefix 2001:DB8:43::/48
Mar 19 12:47:29.751 IST: option CLIENT-LINKLAYER-ADDRESS(79), len 8
Mar 19 12:47:29.751 IST: 001f.c2ad.9d8c
Mar 19 12:47:29.751 IST: option INTERFACE-ID(18), len 7
Mar 19 12:47:29.751 IST: 0x4769302F302F33
Mar 19 12:47:29.751 IST: option REMOTEID(37), len 22
Mar 19 12:47:29.751 IST: 0x000000090200030000000000A000300011CE85DC53880
Mar 19 12:47:29.751 IST: IPv6 DHCP: Using interface pool LPDCONF
Mar 19 12:47:29.751 IST: IPv6 DHCP: Creating binding for FE80::21F:C2FF:FEAD:9D8C in pool
LPDCONF
Mar 19 12:47:29.751 IST: IPv6 DHCP: Allocating IA_PD 00120001 in binding for
FE80::21F:C2FF:FEAD:9D8C
Mar 19 12:47:29.751 IST: IPv6 DHCP: Allocating prefix 2001:DB8:43::/48 in binding for
FE80::21F:C2FF:FEAD:9D8C, IAID 00120001
Mar 19 12:47:29.751 IST: IPv6 DHCP: Added Prefix 2001:DB8:43::/48 to Radix tree
Mar 19 12:47:29.752 IST: IPv6 DHCP: Route added: 2001:DB8:43::/48 via
FE80::1EE8:5DFF:FEC5:3884 dist 1 iaid 00120001 vrf default
Mar 19 12:47:29.752 IST: IPv6 DHCP_AAA: Retrieved subblock; It has AAA DNS_SERVERS=0
Mar 19 12:47:29.752 IST: IPv6 DHCP: Option CLIENT-LINKLAYER-ADDRESS(79) in Relay-forward
ignored: 0001001FC2AD9D8C
Mar 19 12:47:29.752 IST: IPv6 DHCP: Option REMOTEID(37):
000000090200030000000000A000300011CE85DC53880
Mar 19 12:47:29.752 IST: IPv6 DHCP: SAS retured Null falling to link local
Mar 19 12:47:29.752 IST: IPv6 DHCP: Returning Link local address FE80::F61F:C2FF:FEAD:A0A8
```

```
Mar 19 12:47:29.752 IST: IPv6 DHCP: detailed packet contents
Mar 19 12:47:29.752 IST: src FE80::F61F:C2FF:FEAD:A0A8
Mar 19 12:47:29.752 IST: dst FE80::1EE8:5DFF:FEC5:3884 (GigabitEthernet0/4)
Mar 19 12:47:29.752 IST: type RELAY-REPLY(13), hop 0
Mar 19 12:47:29.752 IST: link ::
Mar 19 12:47:29.752 IST: peer FE80::21F:C2FF:FEAD:9D8C
Mar 19 12:47:29.752 IST: option RELAY-MSG(9), len 111
Mar 19 12:47:29.752 IST: type REPLY(7), xid 1606592
Mar 19 12:47:29.752 IST: option SERVERID(2), len 10
Mar 19 12:47:29.752 IST: 00030001F41FC2ADA080
Mar 19 12:47:29.752 IST: option CLIENTID(1), len 10
Mar 19 12:47:29.752 IST: 00030001001FC2AD9D80
Mar 19 12:47:29.752 IST: option IA-PD(25), len 41
Mar 19 12:47:29.752 IST: IAID 0x00120001, T1 300, T2 480
Mar 19 12:47:29.752 IST: option IAPREFIX(26), len 25
Mar 19 12:47:29.752 IST: preferred 600, valid 1800, prefix 2001:DB8:43::/48
Mar 19 12:47:29.752 IST: option DNS-SERVERS(23), len 16
Mar 19 12:47:29.752 IST: 2001:DB8::58
Mar 19 12:47:29.752 IST: option DOMAIN-LIST(24), len 10
Mar 19 12:47:29.752 IST: nour.com
Mar 19 12:47:29.752 IST: option INTERFACE-ID(18), len 7
Mar 19 12:47:29.752 IST: 0x4769302F302F33
Mar 19 12:47:29.752 IST: IPv6 DHCP: Sending RELAY-REPLY to FE80::1EE8:5DFF:FEC5:3884 src
FE80::F61F:C2FF:FEAD:A0A8 on GigabitEthernet0/4
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

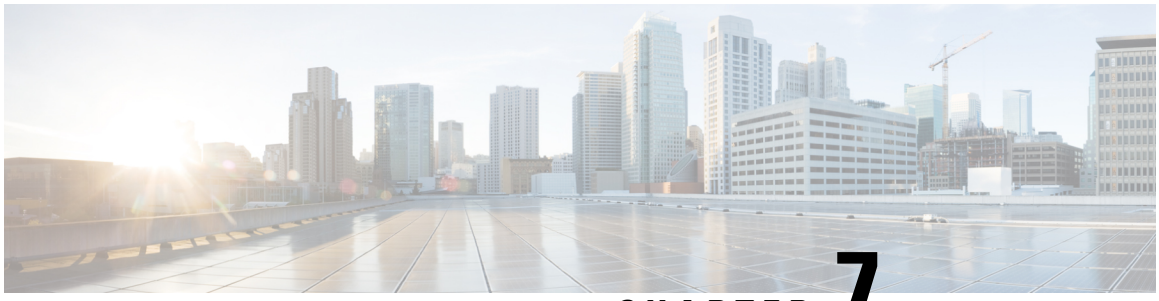
Feature Information for IPv6 Access Services: DHCPv6 Prefix Delegation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 8: Feature Information for IPv6 Access Services: DHCPv6 Prefix Delegation

Feature Name	Releases	Feature Information
IPv6 Access Services: DHCPv6 Prefix Delegation	Cisco IOS XE Release 3.15.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D, ASR-920-10SZ-PD, ASR-920-24SZ-IM, ASR-920-24SZ-M, ASR-920-24TZ-M).



CHAPTER 7

Configuring DHCP Features

This chapter describes how to configure DHCP snooping and option-82 data insertion on the Cisco ASR 920 Series Router.

Limitations and Restrictions

The following limitations and restrictions apply when configuring DHCP features on the Cisco ASR 920 Series router:

- The **ip dhcp snooping binding** command is not supported.
- The DHCPv4 snooping override functionality is not supported.
- DHCP smart relay supports a maximum of 16 local addresses configured on a BDI or an interface.

DHCP Features

DHCP is widely used in LAN environments to dynamically assign host IP addresses from a centralized server, which significantly reduces the overhead of administration of IP addresses. DHCP also helps conserve the limited IP address space because IP addresses no longer need to be permanently assigned to hosts; only those hosts that are connected to the network consume IP addresses.

DHCP Server

The DHCP server assigns IP addresses from specified address pools on a switch or router to DHCP clients and manages them. If the DHCP server cannot give the DHCP client the requested configuration parameters from its database, it can forward the request to one or more secondary DHCP servers defined by the network administrator.

DHCP Relay Agent

A DHCP relay agent is a Layer 3 device that forwards DHCP packets between clients and servers. Relay agents forward requests and replies between clients and servers when they are not on the same physical subnet. Relay agent forwarding is different from the normal Layer 2 forwarding, in which IP datagrams are switched transparently between networks. Relay agents receive DHCP messages and generate new DHCP messages to send on egress interfaces.

DHCP relay is supported on the following routers:

- ASR-920-4SZ-D
- ASR-920-4SZ-A
- ASR-920-10SZ
- ASR-920-12CZ-A
- ASR-920-12CZ-D
- ASR-920-24SZ-M
- ASR-920-24TZ-M
- ASR-920-12SZ-IM
- ASR-920-24SZ-IM

Note DHCP option-82 is not supported, when the DHCP relay agent is enabled and by simultaneously disabling the DHCP snooping.

DHCP Snooping

DHCP snooping is a DHCP security feature that provides network security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding database, also referred to as a DHCP snooping binding table. For more information about this database, see the Displaying DHCP Snooping Information, on page 129.

DHCP snooping acts like a firewall between untrusted hosts and DHCP servers. You use DHCP snooping to differentiate between untrusted interfaces connected to the end user and trusted interfaces connected to the DHCP server or another switch.

Note For DHCP snooping to function properly, all DHCP servers must be connected to the router through trusted interfaces.

An untrusted DHCP message is a message that is received from outside the network or firewall. When you use DHCP snooping in a service-provider environment, an untrusted message is sent from a device that is not in the service-provider network, such as a customer's switch. Messages from unknown devices are untrusted because they can be sources of traffic attacks.

The DHCP snooping binding database has the MAC address, the IP address, the lease time, the binding type, the bridge-domain number, and the interface information that corresponds to the local untrusted interfaces of a switch. It does not have information regarding hosts interconnected with a trusted interface.

In a service-provider network, a trusted interface is connected to a port on a device in the same network. An untrusted interface is connected to an untrusted interface in the network or to an interface on a device that is not in the network.

When a switch receives a packet on an untrusted interface and the interface belongs to a bridge-domain in which DHCP snooping is enabled, the switch compares the source MAC address and the DHCP client hardware address. If the addresses match (the default), the switch forwards the packet. If the addresses do not match, the switch drops the packet.

The router drops a DHCP packet when one of these situations occurs:

- A packet from a DHCP server, such as a DHCPOFFER, DHCPACK, DHCPNAK, or DHCPLEASEQUERY packet, is received from outside the network or firewall.
- A packet is received on an untrusted interface, and the source MAC address and the DHCP client hardware address do not match.
- The switch receives a DHCPRELEASE or DHCPDECLINE broadcast message that has a MAC address in the DHCP snooping binding database, but the interface information in the binding database does not match the interface on which the message was received.
- A DHCP relay agent forwards a DHCP packet that includes a relay-agent IP address that is not 0.0.0.0, or the relay agent forwards a packet that includes option-82 information to an untrusted port.

If the switch is an aggregation switch supporting DHCP snooping and is connected to an edge switch that is inserting DHCP option-82 information, the switch drops packets with option-82 information when packets are received on an untrusted interface. If DHCP snooping is enabled and packets are received on a trusted port, the aggregation switch does not learn the DHCP snooping bindings for connected devices and cannot build a complete DHCP snooping binding database.

When an aggregation switch can be connected to an edge switch through an untrusted interface and you enter the ip dhcp snooping information option allowed-trust global configuration command, the aggregation switch accepts packets with option-82 information from the edge switch. The aggregation switch learns the bindings for hosts connected through an untrusted switch interface. The DHCP security features, such as dynamic ARP inspection, can still be enabled on the aggregation switch while the switch receives packets with option-82 information on ingress untrusted interfaces to which hosts are connected. The port on the edge switch that connects to the aggregation switch must be configured as a trusted interface.

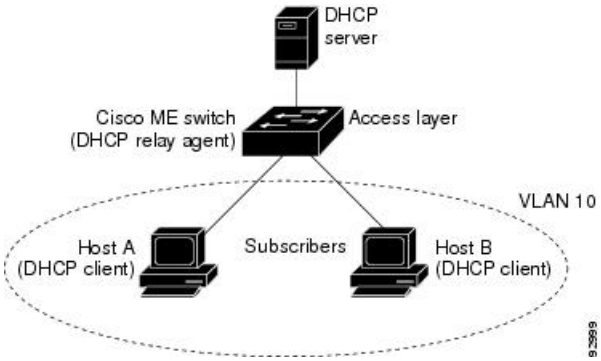
Option-82 Data Insertion

In residential, metropolitan Ethernet-access environments, DHCP can centrally manage the IP address assignments for a large number of subscribers. When the DHCP option-82 feature is enabled on the switch, a subscriber device is identified by the switch port through which it connects to the network (in addition to its MAC address). Multiple hosts on the subscriber LAN can be connected to the same port on the access switch and are uniquely identified.

Note The DHCP option-82 feature is supported only when DHCP snooping is globally enabled and on the bridge-domains to which subscriber devices using this feature are assigned.

Figure below is an example of a metropolitan Ethernet network in which a centralized DHCP server assigns IP addresses to subscribers connected to the switch at the access layer. Because the DHCP clients and their associated DHCP server do not reside on the same IP network or subnet, a DHCP relay agent (the Cisco ASR 920 Series Router) is configured with a helper address to enable broadcast forwarding and to transfer DHCP messages between the clients and the server.

Figure 7: DHCP Relay Agent in a Metropolitan Ethernet Network



When you enable the DHCP snooping information option 82 on the switch, this sequence of events occurs:

- The host (DHCP client) generates a DHCP request and broadcasts it on the network.
- When the switch receives the DHCP request, it adds the option-82 information in the packet. By default, the remote-ID suboption is the switch MAC address, and the circuit-ID suboption is the port identifier from which the packet is received. You can also configure the remote ID and circuit ID. For information on configuring these suboptions, see the Enabling DHCP Snooping and Option 82, on page 125.
- If the IP address of the relay agent is configured, the switch adds this IP address in the DHCP packet.
- The switch forwards the DHCP request that includes the option-82 field to the DHCP server.
- The DHCP server receives the packet. If the server is option-82-capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. Then the DHCP server echoes the option-82 field in the DHCP reply.
- The DHCP server unicasts the reply to the switch if the request was relayed to the server by the switch. The switch verifies that it originally inserted the option-82 data by inspecting the remote ID and possibly the circuit ID fields. The switch removes the option-82 field and forwards the packet to the switch port that connects to the DHCP client that sent the DHCP request.

In the default suboption configuration, when the described sequence of events occurs, the values in these fields in figure below do not change:

- Circuit ID suboption fields
- Suboption type
- Length of the suboption type
- Circuit ID type
- Length of the circuit ID type
- Remote ID suboption fields

- Suboption type
- Length of the suboption type
- Remote ID type
- Length of the circuit ID type

Figure below shows the packet formats for the remote ID suboption and the circuit ID suboption when the default suboption configuration is used. The switch uses the packet formats when DHCP snooping is globally enabled and when the ip dhcp snooping information option global configuration command is entered.

Figure 8: Suboption Packet Formats

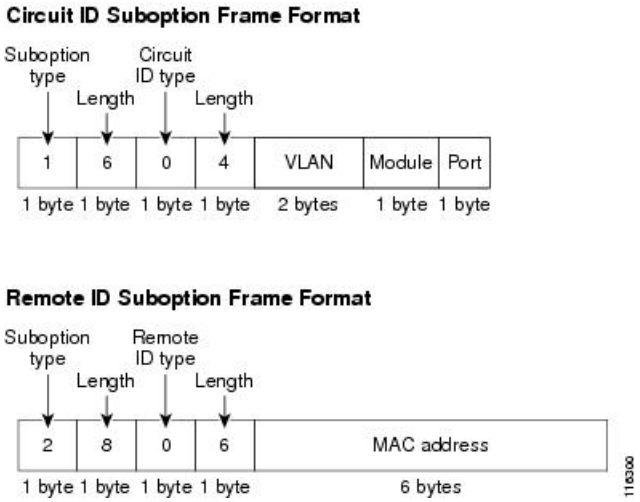


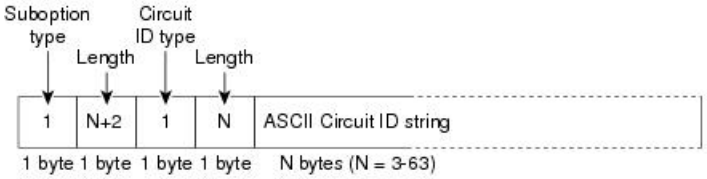
Figure below shows the packet formats for user-configured remote-ID and circuit-ID suboptions The switch uses these packet formats when you globally enable DHCP snooping and enter the ip dhcp snooping information option format remote-id global configuration command and the ip dhcp snooping bridge-domain information option format-type circuit-id string interface configuration command.

The values for these fields in the packets change from the default values when you configure the remote-ID and circuit-ID suboptions:

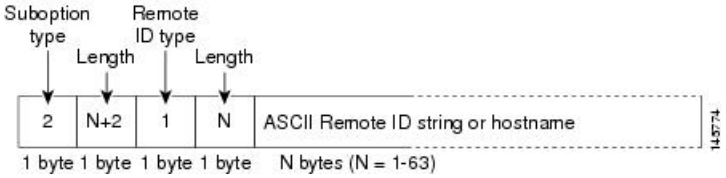
- Circuit-ID suboption fields
- The circuit-ID type is 1.
- The length values are variable, depending on the length of the string that you configure.
- Remote-ID suboption fields
- The remote-ID type is 1.
- The length values are variable, depending on the length of the string that you configure.

Figure 9: User-Configured Suboption Packet Formats

Circuit ID Suboption Frame Format (for user-configured string):



Remote ID Suboption Frame Format (for user-configured string):



Cisco IOS DHCP Server Database

During the DHCP-based autoconfiguration process, the designated DHCP server uses the Cisco IOS DHCP server database. It has IP addresses, address bindings, and configuration parameters, such as the boot file.

An address binding is a mapping between an IP address and a MAC address of a host in the Cisco IOS DHCP server database. You can manually assign the client IP address, or the DHCP server can allocate an IP address from a DHCP address pool. For more information about manual and automatic address bindings, see the IP Addressing Configuration Guide Library, Cisco IOS XE Release 3S .

DHCP Snooping Binding Database

When DHCP snooping is enabled, the switch uses the DHCP snooping binding database to store information about untrusted interfaces. The database can have up to 8192 bindings.

Note DHCP snooping database read event will **not** retrieve entries for 10G and PC interface

Each database entry (*binding*) has an IP address, an associated MAC address, the lease time (in hexadecimal format), the interface to which the binding applies, and the bridge-domain to which the interface belongs. The database agent stores the bindings in a file at a configured location. At the end of each entry is a *checksum* value that accounts for all the bytes associated with the entry. Each entry is 72 bytes, followed by a space and then the checksum value.

To keep the bindings when the switch reloads, you must use the DHCP snooping database agent. If the agent is disabled, dynamic ARP inspection, and the DHCP snooping binding database has dynamic bindings, the switch loses its connectivity. If the agent is disabled and only DHCP snooping is enabled, the switch does not lose its connectivity, but DHCP snooping might not prevent DHCP spoofing attacks.

When reloading, the router reads the binding file to build the DHCP snooping binding database. The switch keeps the file current by updating it when the database changes.

When a router learns of new bindings or when it loses bindings, the router immediately updates the entries in the database. The router also updates the entries in the binding file. The frequency at which the file is updated is based on a configurable delay, and the updates are batched. If the file is not updated in a specified time (set by the write-delay and abort-timeout values), the update stops.

This is the format of the file that has the bindings:

```
<initial-checksum>

TYPE DHCP-SNOOPING

VERSION 1

BEGIN

<entry-1> <checksum-1>

<entry-2> <checksum-1-2>

...

...

<entry-n> <checksum-1-2-..-n>

END
```

Each entry in the file is tagged with a checksum value that the router uses to verify the entries when it reads the file. The *initial-checksum* entry on the first line distinguishes entries associated with the latest file update from entries associated with a previous file update.

This is an example of a binding file:

```
2bb4c2a1

TYPE DHCP-SNOOPING

VERSION 1

BEGIN

192.1.168.1 3 0003.47d8.c91f 2BB6488E Fa1/0/4 21ae5fbb

192.1.168.3 3 0003.44d6.c52f 2BB648EB Fa1/0/4 1bdb223f

192.1.168.2 3 0003.47d9.c8f1 2BB648AB Fa1/0/4 584a38f0

END
```

When the router starts and the calculated checksum value equals the stored checksum value, the router reads entries from the binding file and adds the bindings to its DHCP snooping binding database. The router ignores an entry when one of these situations occurs:

- The router reads the entry and the calculated checksum value does not equal the stored checksum value. The entry and the ones following it are ignored.
- An entry has an expired lease time (the router might not remove a binding entry when the lease time expires).
- The interface in the entry no longer exists on the system.
- The interface is a routed interface or a DHCP snooping-trusted interface.

Configuring DHCP Features

Default DHCP Configuration

Table below shows the default DHCP configuration.

Table 9: Default DHCP Configuration

Feature	Default Setting
DHCP server	Enabled in Cisco IOS software, requires configuration[1]
DHCP relay agent	Enabled[2]
DHCP packet forwarding address	None configured
Checking the relay agent information	Enabled (invalid messages are dropped)[3]
DHCP relay agent forwarding policy	Replace the existing relay agent information[4]
DHCP snooping enabled globally	Disabled
DHCP snooping information option	Enabled
DHCP snooping option to accept packets on untrusted ingress interfaces[5]	Disabled
DHCP snooping limit rate	None configured
DHCP snooping trust	Untrusted
DHCP snooping bridge-domain	Disabled
DHCP snooping MAC address verification	Enabled
Cisco IOS DHCP server binding database	Enabled in Cisco IOS software, requires configuration. **Note** The router gets network addresses and configuration parameters only from a device configured as a DHCP server.
DHCP snooping binding database agent	Enabled in Cisco IOS software, requires configuration. This feature is operational only when a destination is configured.

1 The router responds to DHCP requests only if it is configured as a DHCP server.
2 The router relays DHCP packets only if the IP address of the DHCP server is configured on the BDI of the DHCP client.
3 The router relays DHCP packets only if the IP address of the DHCP server is configured on the BDI of the DHCP client.
4 The router relays DHCP packets only if the IP address of the DHCP server is configured on the BDI of the DHCP client.
5 Use this feature when the router is an aggregation switch that receives packets with option-82 information from an edge switch.

DHCP Snooping Configuration Guidelines

- You must globally enable DHCP snooping on the router.
- DHCP snooping is not active until DHCP snooping is enabled on a bridge-domain.
- Before globally enabling DHCP snooping on the router, make sure that the devices acting as the DHCP server and the DHCP relay agent are configured and enabled.
- Before configuring the DHCP snooping information option on your router, be sure to configure the device that is acting as the DHCP server. For example, you must specify the IP addresses that the DHCP server can assign or exclude, or you must configure DHCP options for these devices.
- When configuring a large number of circuit IDs on a router, consider the impact of lengthy character strings on the NVRAM or the flash memory. If the circuit-ID configurations, combined with other data, exceed the capacity of the NVRAM or the flash memory, an error message appears.
- Before configuring the DHCP relay agent on your router, make sure to configure the device that is acting as the DHCP server. For example, you must specify the IP addresses that the DHCP server can assign or exclude, configure DHCP options for devices, or set up the DHCP database agent.
- If the DHCP relay agent is enabled but DHCP snooping is disabled, the DHCP option-82 data insertion feature is not supported.
- If a switch port is connected to a DHCP server, configure a port as trusted by entering the ip dhcp snooping trust interface configuration command.
- If a switch port is connected to a DHCP client, configure a port as untrusted by entering the no ip dhcp snooping trust interface configuration command.

Follow these guidelines when configuring the DHCP snooping binding database:

- Because both NVRAM and the flash memory have limited storage capacity, we recommend that you store the binding file on a TFTP server.
- For network-based URLs (such as TFTP and FTP), you must create an empty file at the configured URL before the router can write bindings to the binding file at that URL. See the documentation for your TFTP server to determine whether you must first create an empty file on the server; some TFTP servers cannot be configured this way.
- To ensure that the lease time in the database is accurate, we recommend that NTP is enabled and configured.
- If NTP is configured, the router writes binding changes to the binding file only when the router system clock is synchronized with NTP.
- Do not enter the ip dhcp snooping information option allowed-untrusted command on an aggregation switch to which an untrusted device is connected. If you enter this command, an untrusted device might spoof the option-82 information.

```
Router# show ip dhcp snooping binding

MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface

------------------  ---------------  ----------  -------------  ----  --------------------
```

```
AA:00:11:13:00:01   40.0.0.2         117         dhcp-snooping  100   Port-channel100+Efp7
```

Configuring the DHCP Server

The router can act as a DHCP server. By default, the Cisco IOS DHCP server and relay agent features are enabled on your router but are not configured. These features are not operational.

For procedures to configure the router as a DHCP server, see the IP Addressing Configuration Guide Library, Cisco IOS XE Release 3S.

Configuring the DHCP Relay Agent

SUMMARY STEPS

1. **configure terminal**
2. **service dhcp**
3. **end**
4. **show running-config**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**service dhcp**	Enable the DHCP relay agent on your router. By default, this feature is enabled.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show running-config**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

What to do next

To disable the DHCP relay agent, use the **no service dhcp** global configuration command.

Specifying the Packet Forwarding Address

If the DHCP server and the DHCP clients are on different networks or subnets and the router is running the metro IP access image, you must configure the router with the **ip helper-address** *address* interface configuration command. The general rule is to configure the command on the Layer 3 interface closest to the client. The address used in the **ip helper-address** command can be a specific DHCP server IP address, or it can be the network address if other DHCP servers are on the destination network segment. Using the network address enables any DHCP server to respond to requests.

Note To remove the DHCP packet forwarding address, use the **no ip helper-address***address* interface configuration command.

Beginning in privileged EXEC mode, follow these steps to specify the packet forwarding address:

SUMMARY STEPS

1. **configure terminal**
2. **interface bridge-domain id**
3. **ip address ip-address subnet-mask**
4. **ip helper-address address**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface bridge-domain id**	Create a switch virtual interface by entering a bridge-domain ID, and enter interface configuration mode.
Step 3	**ip address ip-address subnet-mask**	Configure the interface with an IP address and an IP subnet.
Step 4	**ip helper-address address**	Specify the DHCP packet forwarding address. The helper address can be a specific DHCP server address, or it can be the network address if other DHCP servers are on the destination network segment. Using the network address enables other servers to respond to DHCP requests. If you have multiple servers, you can configure one helper address for each server.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show running-config**	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Enabling DHCP Snooping and Option 82

Beginning in the privileged EXEC mode, follow these steps to enable DHCP snooping on the switch:

SUMMARY STEPS

1. **configure terminal**
2. **ip dhcp snooping**

3. **ip dhcp snooping bridge-domain id**
4. **ip dhcp snooping information option**
5. **ip dhcp snooping information option format remote-id [string ASCII-string | hostname]**
6. **ip dhcp snooping information option allowed-untrusted**
7. **interface interface-id**
8. **no shutdown**
9. **ip dhcp snooping bridge-domain id information option format-type circuit-id string ASCII-string**
10. **ip dhcp snooping trust**
11. **ip dhcp snooping limit rate rate**
12. **exit**
13. **ip dhcp snooping verify mac-address**
14. **end**
15. **show running-config**
16. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip dhcp snooping**	Enable DHCP snooping globally.
Step 3	**ip dhcp snooping bridge-domain id**	Enable DHCP snooping on a bridge-domain
Step 4	**ip dhcp snooping information option**	Enable the router to insert and remove DHCP relay information (option-82 field) in forwarded DHCP request messages to the DHCP server. This is the default setting.
Step 5	**ip dhcp snooping information option format remote-id [string ASCII-string | hostname]**	(Optional) Configure the remote-ID suboption. You can configure the remote ID to be: • String of up to 63 ASCII characters (no spaces) • Configured hostname for the router **Note** If the hostname is longer than 63 characters, it is truncated to 63 characters in the remote-ID configuration. The default remote ID is the router MAC address.
Step 6	**ip dhcp snooping information option allowed-untrusted**	(Optional) If the router is acting as an aggregation switch connected to an edge switch, enable the router to accept incoming DHCP snooping packets with option-82 information from the edge switch. The default is disabled. **Note** Enter this command only on aggregation switches that are connected to trusted devices.

	Command or Action	Purpose
Step 7	**interface interface-id**	Specify the interface to be configured, and enter interface configuration mode.
Step 8	**no shutdown**	Enable the port, if necessary. By default, UNIs and ENIs are disabled and NNIs are enabled.
Step 9	**ip dhcp snooping bridge-domain id information option format-type circuit-id string ASCII-string**	(Optional) Configure the circuit-ID suboption for the specified interface. The default circuit ID is the port identifier. You can configure the circuit ID to be a string of 3 to 63 ASCII characters (no spaces).
Step 10	**ip dhcp snooping trust**	Configure the interface as trusted or untrusted. You can use the **no** keyword to configure an interface to receive messages from an untrusted client. The default is untrusted.
Step 11	**ip dhcp snooping limit rate rate**	(Optional) Configure the number of DHCP packets per second that an interface can receive. The range is 1 to 2048. By default, no rate limit is configured. **Note** We recommend an untrusted rate limit of not more than 100 packets per second. If you configure rate limiting for trusted interfaces, you might need to increase the rate limit if the port is a trunk port assigned to more than one bridge-domain on which DHCP snooping is enabled.
Step 12	**exit**	Return to global configuration mode.
Step 13	**ip dhcp snooping verify mac-address**	(Optional) Configure the router to verify that the source MAC address in a DHCP packet that is received on untrusted ports matches the client hardware address in the packet. The default is to verify that the source MAC address matches the client hardware address in the packet.
Step 14	**end**	Return to privileged EXEC mode.
Step 15	**show running-config**	Verify your entries.
Step 16	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Enabling the Cisco IOS DHCP Server Database

For procedures to enable and configure the Cisco IOS DHCP server database, see the *IP Addressing: DHCP Configuration Guide*.

Enabling the DHCP Snooping Binding Database Agent

Beginning in privileged EXEC mode, follow these steps to enable and configure the DHCP snooping binding database agent on the router:

SUMMARY STEPS

1. **configure terminal**
2. **ip dhcp snooping database {flash:/filename | ftp://user:password@host/filename | http://[[username:password]@]{hostname | host-ip}[/directory] /image-name.tar | rcp://user@host/filename}| tftp://host/filename**
3. **ip dhcp snooping database timeout seconds**
4. **ip dhcp snooping database write-delay seconds**
5. **end**
6. **ip dhcp snooping binding [ip-address | mac-address | dynamic | static | bridge-domain id | interface interface]**
7. **show ip dhcp snooping database [detail]**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip dhcp snooping database {flash:/filename | ftp://user:password@host/filename | http://[[username:password]@]{hostname | host-ip}[/directory] /image-name.tar | rcp://user@host/filename}| tftp://host/filename**	Specify the URL for the database agent or the binding file by using one of these forms: • **flash:**/*filename* • **ftp:**//*user:password@host/filename* • **http:**//[[*username:password]@]{hostname | host-ip}[/directory]/image-name***.tar*** • **rcp:**//*user@host/filename* • **tftp:**//*host/filename*
Step 3	**ip dhcp snooping database timeout seconds**	Specify when to stop the database transfer process after the binding database changes. The range is from 0 to 86400. Use 0 for an infinite duration. The default is 300 seconds (5 minutes).
Step 4	**ip dhcp snooping database write-delay seconds**	Specify the duration for which the transfer should be delayed after the binding database changes. The range is from 15 to 86400 seconds. The default is 300 seconds (5 minutes).
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**ip dhcp snooping binding [ip-address | mac-address | dynamic | static | bridge-domain id | interface interface]**	(Optional) Add binding entries to the DHCP snooping binding database. The *seconds* range is from 1 to 4294967295. Enter this command for each entry that you add. **Note** Use this command when you are testing or debugging the router.

	Command or Action	Purpose
Step 7	**show ip dhcp snooping database [detail]**	Display the status and statistics of the DHCP snooping binding database agent.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Stopping the Database Agent and Binding files

To stop using the database agent and binding files, use the no ip dhcp snooping database global configuration command.To reset the timeout or delay values, use the ip dhcp snooping database timeout seconds or the ip dhcp snooping database write-delay seconds global configuration command.

Clearing the Statistics of the DHCP Snooping Binding Database Agent

To clear the statistics of the DHCP snooping binding database agent, use the clear ip dhcp snooping database statistics privileged EXEC command. To renew the database, use the renew ip dhcp snooping database privileged EXEC command.

Deleting Binding Entries from the DHCP Snooping Binding Database

To delete binding entries from the DHCP snooping binding database, use the no ip dhcp snooping binding mac-address bridge-domain id ip-address interface interface-id privileged EXEC command. Enter this command for each entry that you delete.

Disabling DHCP Snooping

To disable DHCP snooping, use the **no ip dhcp snooping** global configurationcommand.

To disable DHCP snooping on a bridge-domain, use the **no ip dhcp snooping bridge-domain** *id* global configuration command.

To disable the insertion and removal of the option-82 field, use the **no ip dhcp snooping information option global** configuration command.

To configure an aggregation switch to drop incoming DHCP snooping packets with option-82 information from an edge switch, use the **no ip dhcp snooping information option allowed-untrusted** global configuration command.

Displaying DHCP Snooping Information

To display the DHCP snooping information, use one or more of the privileged EXEC commands as shown in below table:

Table 10: Commands for Displaying DHCP Information

Command	Purpose
show ip dhcp snooping	Displays the DHCP snooping configuration

Command	Purpose
show ip dhcp snooping binding*[ip-address | mac-address* | **dynamic | static | bridge-domain** *id* | **interface***interface]*	Displays only the dynamically configured bindings in the DHCP snooping binding database, also referred to as a binding table.[6]
show ip dhcp snooping database	Displays the DHCP snooping binding database status and statistics.

[6] If DHCP snooping is enabled and an interface changes to the down state, the router does not delete the manually configured bindings.

Pre-assigned Address Reserved in the DHCP Pool

This example shows that the preassigned address was correctly reserved in the DHCP pool:

```
Router# show ip dhcp pool dhcppool
Pool dhcp pool:
 Utilization mark (high/low) : 100 / 0
 Subnet size (first/next) : 0 / 0
 Total addresses : 254
 Leased addresses : 0
 Excluded addresses : 4
 Pending event : none
 1 subnet is currently in the pool:
 Current index   IP address range         Leased/Excluded/Total
 10.1.1.1        10.1.1.1 - 10.1.1.254     0     / 4 / 254
 1 reserved address is currently in the pool
 Address         Client
 10.1.1.7 Et1/0
```

Automatic Generation of Subscriber Identifier

In this example, a subscriber identifier is automatically generated, and the DHCP server ignores any client identifier fields in the DHCP messages and uses the subscriber identifier instead. The subscriber identifier is based on the short name of the interface and the client preassigned IP address 10.1.1.7.

```
Router# show running config
Building configuration...
Current configuration : 4899 bytes
!
hostname router
!
no aaa new-model
clock timezone EST 0
ip subnet-zero
ip dhcp relay information policy removal pad
no ip dhcp use vrf connected
ip dhcp use subscriber-id client-id
ip dhcp subscriber-id interface-name
ip dhcp excluded-address 10.1.1.1 10.1.1.3
!
ip dhcp pool dhcppool
 network 10.1.1.0 255.255.255.0
 address 10.1.1.7 client-id "Et1/0" ascii
#output truncated#
```

Additional References

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

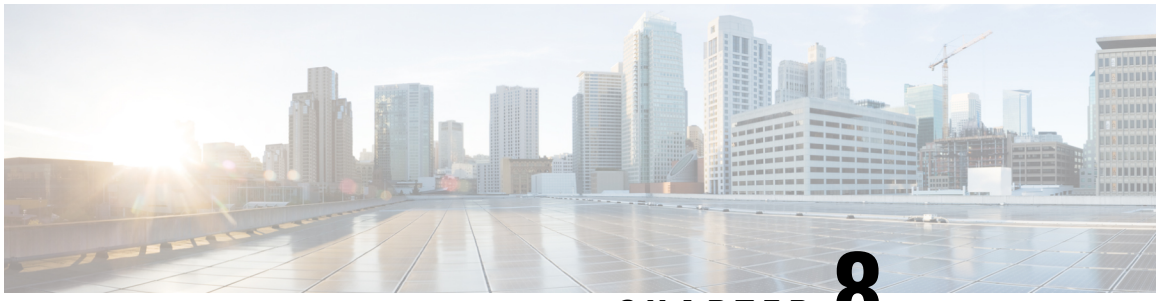
Feature Information for Configuring DHCP Features

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11: Feature Information for DHCP Features

Feature Name	Releases	Feature Configuration Information
Configuring DHCP Features	Cisco IOS XE Release 3.13.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D).



CHAPTER 8

Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol inspection (dynamic ARP inspection). This feature helps prevent malicious attacks on the router by not relaying invalid ARP requests and responses to other bridge-domains.

Note For complete syntax and usage information for the commands used in this chapter, see the command reference for this release.

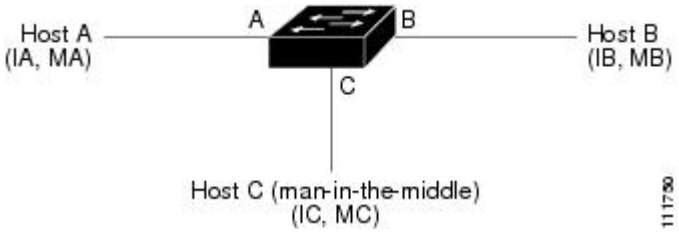
Note The Cisco ASR 920 Series Router supports dynamic ARP inspection only on bridge-domains; other interfaces such as VLANs are not supported.

Dynamic ARP Inspection

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, Host B wants to send information to Host A but does not have the MAC address of Host A in its ARP cache. Host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of Host A. All hosts within the broadcast domain receive the ARP request, and Host A responds with its MAC address. However, because ARP allows a gratuitous reply from a host even if an ARP request was not received, an ARP spoofing attack and the poisoning of ARP caches can occur. After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

A malicious user can attack hosts, switches, and routers connected to your Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet. Figure below shows an example of ARP cache poisoning.

Figure 10: ARP Cache Poisoning



Hosts A, B, and C are connected to the switch on interfaces A, B and C, all of which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, Host A uses IP address IA and MAC address MA. When Host A needs to communicate to Host B at the IP layer, it broadcasts an ARP request for the MAC address associated with IP address IB. When the switch and Host B receive the ARP request, they populate their ARP caches with an ARP binding for a host with the IP address IA and a MAC address MA; for example, IP address IA is bound to MAC address MA. When Host B responds, the switch and Host A populate their ARP caches with a binding for a host with the IP address IB and the MAC address MB.

Host C can poison the ARP caches of the switch, Host A, and Host B by broadcasting forged ARP responses with bindings for a host with an IP address of IA (or IB) and a MAC address of MC. Hosts with poisoned ARP caches use the MAC address MC as the destination MAC address for traffic intended for IA or IB. This means that Host C intercepts that traffic. Because Host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. Host C has inserted itself into the traffic stream from Host A to Host B, the classic *man-in-the middle* attack.

Dynamic ARP inspection is a security feature that validates ARP packets in a network. It intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. This capability protects the network from certain man-in-the-middle attacks.

Dynamic ARP inspection ensures that only valid ARP requests and responses are relayed. The router performs these activities:

- Intercepts all ARP requests and responses on untrusted ports
- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

Dynamic ARP inspection determines the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a trusted database, the DHCP snooping binding database. This database is built by DHCP snooping if DHCP snooping is enabled on the bridge-domains and on the router. If the ARP packet is received on a trusted interface, the router forwards the packet without any checks. On untrusted interfaces, the switch forwards the packet only if it is valid.

You enable dynamic ARP inspection on a per-bridge-domain basis by using the **ip arp inspection bridge-domain** domain-id global configuration command.

In non-DHCP environments, dynamic ARP inspection can validate ARP packets against user-configured ARP access control lists (ACLs) for hosts with statically configured IP addresses. You define an ARP ACL by using the arp access-list *acl-name* global configuration command. For configuration information, see the Configuring Dynamic ARP Inspection in DHCP Environments. The switch logs dropped packets. For more information about the log buffer, see the Configuring the Log Buffer (optional).

You can configure dynamic ARP inspection to drop ARP packets when the IP addresses in the packets are invalid or when the MAC addresses in the body of the ARP packets do not match the addresses specified in

the Ethernet header. Use the **ip arp inspection validate** [**src-mac**] [**dst-mac**] [**ip**] global configuration command. For more information, see the Performing Validation Checks (optional).

Interface Trust States and Network Security

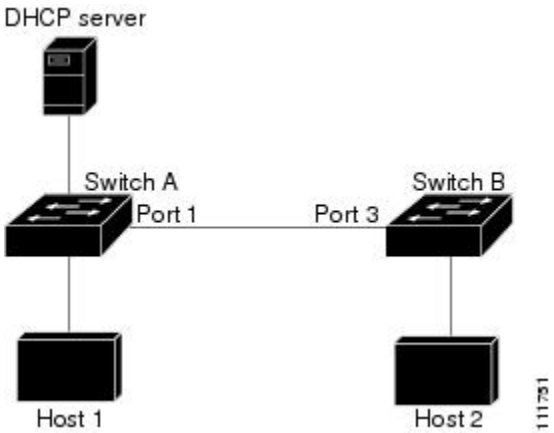
Dynamic ARP inspection associates a trust state with each interface on the router. Packets arriving on trusted interfaces bypass all dynamic ARP inspection validation checks, and those arriving on untrusted interfaces undergo the dynamic ARP inspection validation process.

In a typical network configuration, you configure all switch ports connected to host ports as untrusted and configure all switch ports connected to switches as trusted. With this configuration, all ARP packets entering the network from a given switch bypass the security check. No other validation is needed at any other place in the bridge-domain or in the network. You configure the trust setting by using the **ip arp inspection trust interface** configuration command.

Note Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

In the figure below, assume that both Switch A and Switch B are running dynamic ARP inspection on the bridge-domain that includes Host 1 and Host 2. If Host 1 and Host 2 acquire their IP addresses from the DHCP server connected to Switch A, only Switch A binds the IP-to-MAC address of Host 1. Therefore, if the interface between Switch A and Switch B is untrusted, the ARP packets from Host 1 are dropped by Switch B. Connectivity between Host 1 and Host 2 is lost.

Figure 11: ARP Packet Validation on a Bridge-Domain Enabled for Dynamic ARP Inspection



Configuring interfaces to be trusted when they are actually untrusted leaves a security hole in the network. If Switch A is not running dynamic ARP inspection, Host 1 can easily poison the ARP cache of Switch B (and Host 2, if the link between the switches is configured as trusted). This condition can occur even though Switch B is running dynamic ARP inspection.

Dynamic ARP inspection ensures that hosts (on untrusted interfaces) connected to a switch running dynamic ARP inspection do not poison the ARP caches of other hosts in the network. However, dynamic ARP inspection does not prevent hosts in other portions of the network from poisoning the caches of the hosts that are connected to a switch running dynamic ARP inspection.

Note Depending on the setup of the DHCP server and the network, it might not be possible to validate a given ARP packet on all switches in the bridge-domain.

Rate Limiting of ARP Packets

The switch CPU performs Dynamic ARP Inspection validation checks; therefore, the number of incoming ARP packets is rate-limited to prevent a denial-of-service attack. By default, the rate for untrusted interfaces is 15 packets per second (pps). Trusted interfaces are not rate-limited. You can change this setting by using the ip arp inspection limit interface configuration command.

Note You can configure a maximum of 1150 packets per second using the **ip arp inspection limit rate** command, although the range specified in the command is 0–2048 packets per second.

For configuration information, see the Limiting the Rate of Incoming ARP Packets (optional).

Relative Priority of ARP ACLs and DHCP Snooping Entries

Dynamic ARP inspection uses the DHCP snooping binding database for the list of valid IP-to-MAC address bindings.

ARP ACLs take precedence over entries in the DHCP snooping binding database. The switch uses ACLs only if you configure them by using the ip arp inspection filter bridge-domain global configuration command. The switch first compares ARP packets to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, the switch also denies the packet even if a valid binding exists in the database populated by DHCP snooping.

Logging of Dropped Packets

When the switch drops a packet, it places an entry in the log buffer and then generates system messages on a rate-controlled basis. After the message is generated, the switch clears the entry from the log buffer. Each log entry contains flow information, such as the receiving bridge-domain, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

You use the ip arp inspection log-buffer global configuration command to configure the number of entries in the buffer and the number of entries needed in the specified interval to generate system messages. You specify the type of packets that are logged by using the ip arp inspection bridge-domain logging global configuration command. For configuration information, see the . Configuring the Log Buffer (optional), on page 146.

Configuring Dynamic ARP Inspection

Default Dynamic ARP Inspection Configuration

Table below shows the default dynamic ARP inspection configuration.

Table 12: Default Dynamic ARP Inspection Configuration

Feature	Default Setting
Dynamic ARP inspection	Disabled on all bridge-domains.
Interface trust state	All interfaces are untrusted.
Rate limit of incoming ARP packets	The rate is 15 pps on untrusted interfaces, assuming that the network is a switched network with a host connecting to as many as 15 new hosts per second. The rate is unlimited on all trusted interfaces. The burst interval is 1 second.
ARP ACLs for non-DHCP environments	No ARP ACLs are defined.
Validation checks	No checks are performed.
Log buffer	When dynamic ARP inspection is enabled, all denied or dropped ARP packets are logged. The number of entries in the log is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second.
Per-bridge-domain logging	All denied or dropped ARP packets are logged.

Dynamic ARP Inspection Configuration Guidelines

- The Cisco ASR 920 Series Router supports dynamic ARP inspection only on bridge-domains.
- Dynamic ARP inspection is an ingress security feature; it does not perform any egress checking.
- Dynamic ARP inspection is not effective for hosts connected to switches that do not support dynamic ARP inspection or that do not have this feature enabled. Because man-in-the-middle attacks are limited to a single Layer 2 broadcast domain, separate the domain with dynamic ARP inspection checks from the one with no checking. This action secures the ARP caches of hosts in the domain enabled for dynamic ARP inspection.
- Dynamic ARP inspection depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically assigned IP addresses.

Note Configuring **ip arp inspection bridge-domain** *id* command without "ip arp inspection" may impact ARP messages processing via all bridge-domains. To overcome the issue, ensure that **ip arp inspection** command is enabled.

When DHCP snooping is disabled or in non-DHCP environments, use ARP ACLs to permit or to deny packets.

- A physical port can join an EtherChannel port channel only when the trust state of the physical port and the channel port match. Otherwise, the physical port remains suspended in the port channel. A port channel inherits its trust state from the first physical port that joins the channel. Consequently, the trust state of the first physical port need not match the trust state of the channel.

Conversely, when you change the trust state on the port channel, the switch configures a new trust state on all the physical ports that comprise the channel.

- The operating rate for the port channel is cumulative across all the physical ports within the channel. For example, if you configure the port channel with an ARP rate-limit of 400 pps, all the interfaces combined on the channel receive an aggregate 400 pps. The rate of incoming ARP packets on EtherChannel ports is equal to the sum of the incoming rate of packets from all the channel members. Configure the rate limit for EtherChannel ports only after examining the rate of incoming ARP packets on the channel-port members.

The rate of incoming packets on a physical port is checked against the port-channel configuration rather than the physical-ports configuration. The rate-limit configuration on a port channel is independent of the configuration on its physical ports.

If the EtherChannel receives more ARP packets than the configured rate, the channel (including all physical ports) is placed in the error-disabled state.

- Make sure to limit the rate of ARP packets on incoming trunk ports. Configure trunk ports with higher rates to reflect their aggregation and to handle packets across multiple dynamic ARP inspection-enabled bridge-domains. You also can use the ip arp inspection limit none interface configuration command to make the rate unlimited. A high rate-limit on one bridge-domain can cause a denial-of-service attack to other bridge-domains when the software places the port in the error-disabled state.
- When you enable dynamic ARP inspection on the switch, policers that were configured to police ARP traffic are no longer effective. The result is that all ARP traffic is sent to the CPU.

Configuring Dynamic ARP Inspection in DHCP Environments

This procedure shows how to configure dynamic ARP inspection when two switches support this feature. Host 1 is connected to Switch A, and Host 2 is connected to Switch B Both switches are running dynamic ARP inspection on bridge-domain 1 where the hosts are located. A DHCP server is connected to Switch A. Both hosts acquire their IP addresses from the same DHCP server. Therefore, Switch A has the bindings for Host 1 and Host 2, and Switch B has the binding for Host 2.

Note Dynamic ARP inspection depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically assigned IP addresses.

Beginning in privileged EXEC mode, follow these steps to configure dynamic ARP inspection. You must perform this procedure on both switches. This procedure is required.

SUMMARY STEPS

1. **show cdp neighbors**
2. **configure terminal**
3. **ip arp inspection**

4. **ip arp inspection bridge-domain id**
5. **interface interface-id**
6. **no shutdown**
7. **ip arp inspection trust**
8. **end**
9. **show ip arp inspection interfaces show ip arp inspection bridge-domain id**
10. **show ip dhcp snooping binding**
11. **show ip arp inspection statistics bridge-domain id**
12. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show cdp neighbors**	Verify the connection between the switches.
Step 2	**configure terminal**	Enter global configuration mode.
Step 3	**ip arp inspection**	Enables dynamic ARP inspection globally.
Step 4	**ip arp inspection bridge-domain id**	Enable dynamic ARP inspection on a per-bridge-domain basis. By default, dynamic ARP inspection is disabled on all bridge-domains. Specify the same bridge-domain ID for both switches.
Step 5	**interface interface-id**	Specify the interface connected to the other switch, and enter interface configuration mode.
Step 6	**no shutdown**	Enable the port, if necessary. By default, user network interfaces (UNIs) and enhanced network interfaces (ENIs) are disabled, and network node interfaces (NNIs) are enabled.
Step 7	**ip arp inspection trust**	Configure the connection between the switches as trusted. By default, all interfaces are untrusted. The switch does not check ARP packets that it receives from the other switch on the trusted interface. It simply forwards the packets. For untrusted interfaces, the switch intercepts all ARP requests and responses. It verifies that the intercepted packets have valid IP-to-MAC address bindings before updating the local cache and before forwarding the packet to the appropriate destination. The switch drops invalid packets and logs them in the log buffer according to the logging configuration specified with the **ip arp inspection bridge-domain logging** global configuration command. For more information, see the Configuring the Log Buffer (optional), on page 146.
Step 8	**end**	Return to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show ip arp inspection interfaces show ip arp inspection bridge-domain id**	Verify the dynamic ARP inspection configuration.
Step 10	**show ip dhcp snooping binding**	Verify the DHCP bindings.
Step 11	**show ip arp inspection statistics bridge-domain id**	Check the dynamic ARP inspection statistics.
Step 12	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Example for Configuring Dynamic ARP Inspection

This example shows how to configure dynamic ARP inspection on Switch A. You would perform a similar procedure on Switch B:

```
Router(config)# ip arp inspection bridge-domain 1
Router(config)# interface gigabitethernet0/1
Router(config-if)# ip arp inspection trust
```

Disabling Dynamic ARP Inspection

To disable dynamic ARP inspection, use the no ip arp inspection bridge-domain global configuration command.

To return the interfaces to an untrusted state, use the no ip arp inspection trust interface configuration command.

Configuring ARP ACLs for Non-DHCP Environments

This procedure shows how to configure dynamic ARP inspection when Switch B does not support dynamic ARP inspection or DHCP snooping.

If you configure port 1 on Switch A as trusted, a security hole is created because both Switch A and Host 1 could be attacked by either Switch B or Host 2. To prevent this possibility, you must configure port 1 on Switch A as untrusted. To permit ARP packets from Host 2, you must set up an ARP ACL and apply it to bridge-domain 1. If the IP address of Host 2 is not static (it is impossible to apply the ACL configuration on Switch A) you must separate Switch A from Switch B at Layer 3 and use a router to route packets between them.

Beginning in privileged EXEC mode, follow these steps to configure an ARP ACL on Switch A. This procedure is required in non-DHCP environments.

Before you begin

SUMMARY STEPS

1. configure terminal
2. **ip arp inspection**
3. **arp access-list acl-name**
4. **permit ip host sender-ip mac host sender-mac [log]**
5. **exit**
6. **ip arp inspection filter arp-acl-name bridge-domain id [static]**
7. **interface interface-id**

8. **no shutdown**
9. **no ip arp inspection trust**
10. **end**
11. **show arp access-list [acl-name] show ip arp inspection bridge-domain id show ip arp inspection interfaces**
12. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	**ip arp inspection**	Enables dynamic ARP inspection globally.
Step 3	**arp access-list acl-name**	Define an ARP ACL, and enter ARP access-list configuration mode. By default, no ARP access lists are defined. **Note** At the end of the ARP access list, there is an implicit deny ip any mac any command.
Step 4	**permit ip host sender-ip mac host sender-mac [log]**	Permit ARP packets from the specified host (Host 2). • For *sender-ip*, enter the IP address of Host 2. • For *sender-mac*, enter the MAC address of Host 2. • (Optional) Specify **log** to log a packet in the log buffer when it matches the access control entry (ACE). Matches are logged if you also configure the **matchlog** keyword in the **ip arp inspection bridge-domain logging** global configuration command.
Step 5	**exit**	Return to global configuration mode.
Step 6	**ip arp inspection filter arp-acl-name bridge-domain id [static]**	Apply the ARP ACL to the bridge-domain. By default, no defined ARP ACLs are applied to any bridge-domain. • For *arp-acl-name*, specify the name of the ACL created in Step 2. • (Optional) Specify **static** to treat implicit denies in the ARP ACL as explicit denies and to drop packets that do not match any previous clauses in the ACL. DHCP bindings are not used. If you do not specify this keyword, it means that there is no explicit deny in the ACL that denies the packet, and DHCP bindings determine whether a packet is permitted or denied if the packet does not match any clauses in the ACL. ARP packets containing only IP-to-MAC address bindings are compared against the ACL. Packets are permitted only if the access list permits them.

	Command or Action	Purpose
Step 7	**interface interface-id**	Specify the Switch A interface that is connected to Switch B, and enter interface configuration mode.
Step 8	**no shutdown**	Enable the port, if necessary. By default, UNIs and ENIs are disabled, and NNIs are enabled.
Step 9	**no ip arp inspection trust**	Configure the Switch A interface that is connected to Switch B as untrusted. By default, all interfaces are untrusted. For untrusted interfaces, the switch intercepts all ARP requests and responses. It verifies that the intercepted packets have valid IP-to-MAC address bindings before updating the local cache and before forwarding the packet to the appropriate destination. The switch drops invalid packets and logs them in the log buffer according to the logging configuration specified with the **ip arp inspection bridge-domain logging** global configuration command.
Step 10	**end**	Return to privileged EXEC mode.
Step 11	**show arp access-list [acl-name] show ip arp inspection bridge-domain id show ip arp inspection interfaces**	Verify your entries.
Step 12	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Example for Configuring an ARP ACL

This example shows how to configure an ARP ACL called host2 on Switch A, to permit ARP packets from Host 2 (IP address 10.0.0.1 and MAC address 0001.0001.0001), to apply the ACL to bridge-domain 1, and to configure port 1 on Switch A as untrusted:

```
Router(config)# arp access-list host2
Router(config-arp-acl)# permit ip host 10.0.0.1 mac host 1.1.1
Router(config-arp-acl)# exit
Router(config)# ip arp inspection filter host2 bridge-domain 1
Router(config)# interface gigabitethernet0/1
Router(config-if)# no ip arp inspection trust
```

Removing the ARP ACL

To remove the ARP ACL, use the no arp access-list global configuration command. To remove the ARP ACL attached to a bridge-domain, use the no ip arp inspection filter arp-acl-name bridge-domain id global configuration command.

To remove an APR ACL attached to a bridge-domain, use the no ip arp inspection filter arp-acl-name bridge-domain id global configuration command.

Limiting the Rate of Incoming ARP Packets (optional)

The switch CPU performs dynamic ARP inspection validation checks; therefore, the number of incoming ARP packets is rate-limited to prevent a denial-of-service attack.

Note Unless you configure a rate limit on an interface (we recommend 1024 packets), changing the trust state of the interface also changes its rate limit to the default value for that trust state. After you configure the rate limit, the interface retains the rate limit even when its trust state is changed. If you enter the **no ip arp inspection** limit interface configuration command, the interface reverts to its default rate limit.

ARP inspection rate limit will not work for values above 1024.

For configuration guidelines for rate limiting trunk ports and EtherChannel ports, see the Dynamic ARP Inspection Configuration Guidelines.

Beginning in privileged EXEC mode, follow these steps to limit the rate of incoming ARP packets. This procedure is optional.

SUMMARY STEPS

1. **configure terminal**
2. **ip arp inspection**
3. **interface interface-id**
4. **no shutdown**
5. **ip arp inspection limit {rate pps [burst interval seconds] | none}**
6. **exit**
7. **exit**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip arp inspection**	Enables dynamic ARP inspection globally.
Step 3	**interface interface-id**	Specify the interface to be rate-limited, and enter interface configuration mode.
Step 4	**no shutdown**	Enable the port, if necessary. By default, UNIs and ENIs are disabled, and NNIs are enabled.
Step 5	**ip arp inspection limit {rate pps [burst interval seconds] | none}**	Limit the rate of incoming ARP requests and responses on the interface. The default rate is 15 pps on untrusted interfaces and unlimited on trusted interfaces. The burst interval is 1 second. The keywords have these meanings:

	Command or Action	Purpose
		• For **rate** *pps*, specify an upper limit for the number of incoming packets processed per second. The range is 0 to 2048 pps. • (Optional) For **burst interval**seconds, specify the consecutive interval in seconds, over which the interface is monitored for a high rate of ARP packets.The range is 1 to 15. • For **rate none**, specify no upper limit for the rate of incoming ARP packets that can be processed.
Step 6	**exit**	Return to global configuration mode.
Step 7	**exit**	Return to privileged EXEC mode.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file. **Note** To return to the default rate-limit configuration, use the no ip arp inspection limit interface configuration command. To disable error recovery for dynamic ARP inspection, use the no errdisable recovery cause arp-inspection global configuration command.

Performing Validation Checks (optional)

Dynamic ARP inspection intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. You can configure the switch to perform additional checks on the destination MAC address, the sender and target IP addresses, and the source MAC address.

Beginning in privileged EXEC mode, follow these steps to perform specific checks on incoming ARP packets. This procedure is optional.

SUMMARY STEPS

1. **configure terminal**
2. **ip arp inspection**
3. **ip arp inspection validate {[src-mac] [dst-mac] [ip]}**
4. **exit**
5. **show ip arp inspection bridge-domain id**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.

	Command or Action	Purpose
Step 2	**ip arp inspection**	Enables dynamic ARP inspection globally.
Step 3	**ip arp inspection validate {[src-mac] [dst-mac] [ip]}**	Perform a specific check on incoming ARP packets. By default, no checks are performed. The keywords have these meanings: • For **src-mac**, check the source MAC address in the Ethernet header against the sender MAC address in the ARP body. This check is performed on both ARP requests and responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped. • For **dst-mac**, check the destination MAC address in the Ethernet header against the target MAC address in ARP body. This check is performed for ARP responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped. • For **ip**, check the ARP body for invalid and unexpected IP addresses. Addresses include 0.0.0.0, 255.255.255.255, and all IP multicast addresses. Sender IP addresses are checked in all ARP requests and responses, and target IP addresses are checked only in ARP responses. You must specify at least one of the keywords. Each command overrides the configuration of the previous command; that is, if a command enables src and dst mac validations, and a second command enables IP validation only, the src and dst mac validations are disabled as a result of the second command.
Step 4	**exit**	Return to privileged EXEC mode.
Step 5	**show ip arp inspection bridge-domain id**	Verify your settings.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file. **Note** To disable checking, use the no ip arp inspection validate [src-mac] [dst-mac] [ip] global configuration command. To display statistics for forwarded, dropped, and MAC and IP validation failure packets, use the show ip arp inspection statistics privileged EXEC command.

Configuring the Log Buffer (optional)

Note Log buffering is not currently supported.

When the switch drops a packet, it places an entry in the log buffer and then generates system messages on a rate-controlled basis. After the message is generated, the switch clears the entry from the log buffer. Each log entry contains flow information, such as the receiving bridge-domain, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

A log-buffer entry can represent more than one packet. For example, if an interface receives many packets on the same bridge-domain with the same ARP parameters, the switch combines the packets as one entry in the log buffer and generates a single system message for the entry.

If the log buffer overflows, it means that a log event does not fit into the log buffer, and the display for the **show ip arp inspection log** privileged EXEC command is affected. A -- in the display appears in place of all data except the packet count and the time. No other statistics are provided for the entry. If you see this entry in the display, increase the number of entries in the log buffer or increase the logging rate.

Beginning in privileged EXEC mode, follow these steps to configure the log buffer. This procedure is optional.

SUMMARY STEPS

1. **configure terminal**
2. **ip arp inspection log-buffer {entries number | logs number interval seconds}**
3. **ip arp inspection bridge-domain id logging {acl-match {matchlog | none} | dhcp-bindings {all | none | permit}}**
4. **exit**
5. **show ip arp inspection log**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip arp inspection log-buffer {entries number | logs number interval seconds}**	Configure the dynamic ARP inspection logging buffer. By default, when dynamic ARP inspection is enabled, denied or dropped ARP packets are logged. The number of log entries is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second. The keywords have these meanings: • For **entries** number, specify the number of entries to be logged in the buffer. The range is 0 to 1024. • For **logs** number **interval** seconds, specify the number of entries to generate system messages in the specified interval.

	Command or Action	Purpose
		For **logs** number, the range is 0 to 1024. A 0 value means that the entry is placed in the log buffer, but a system message is not generated. For **interval** seconds, the range is 0 to 86400 seconds (1 day). A 0 value means that a system message is immediately generated (and the log buffer is always empty). An interval setting of 0 overrides a log setting of 0. The **logs** and **interval** settings interact. If the **logs** number X is greater than **interval** seconds Y, X divided by Y (X/Y) system messages are sent every second. Otherwise, one system message is sent every Y divided by X (Y/X) seconds.
Step 3	**ip arp inspection bridge-domain id logging {acl-match {matchlog | none} | dhcp-bindings {all | none | permit}}**	Control the type of packets that are logged per bridge-domain. By default, all denied or all dropped packets are logged. The term *logged* means the entry is placed in the log buffer and a system message is generated. The keywords have these meanings: • For **acl-match** matchlog, log packets based on the ACE logging configuration. If you specify the **matchlog** keyword in this command and the **log** keyword in the **permit** or **deny** ARP access-list configuration command, ARP packets permitted or denied by the ACL are logged. • For **acl-match none**, do not log packets that match ACLs. • For dhcp-bindings all, log all packets that match DHCP bindings. • For dhcp-bindings none, do not log packets that match DHCP bindings. • For dhcp-bindings permit, log DHCP-binding permitted packets.
Step 4	**exit**	Return to privileged EXEC mode.
Step 5	**show ip arp inspection log**	Verify your settings.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Returning to the Default Log Buffer Settings

To return to the default log buffer settings, use the no ip arp inspection log-buffer {entries | logs} global configuration command.

To return to the default bridge-domain log settings, use the no ip arp inspection bridge-domain id logging {acl-match | dhcp-bindings} global configuration command.

To clear the log buffer, use the clear ip arp inspection log privileged EXEC command.

Displaying Dynamic ARP Inspection Information

To display dynamic ARP inspection information, use the privileged EXEC commands described in table below.

Table 13: Commands for Displaying Dynamic ARP Inspection Information

Command	Description
show arp access-list [acl-name]	Displays detailed information about ARP ACLs.
show ip arp inspection interfaces [interface-id]	Displays the trust state and the rate limit of ARP packets for the specified interface or all interfaces.
show ip arp inspection bridge-domain id	Displays the configuration and the operating state of dynamic ARP inspection for the specified bridge-domain. If a range is specified, displays information for bridge domains with dynamic ARP inspection enabled (active).

Clearing or Displaying Dynamic ARP Inspection Statistics

To clear or display dynamic ARP inspection statistics, use the privileged EXEC commands in table below.

For the show ip arp inspection statistics command, the switch increments the number of forwarded packets for each ARP request and response packet on a trusted dynamic ARP inspection port. The switch increments the number of ACL orDHCP permitted packets for each packet that is denied by source MAC, destination MAC, or IP validation checks, and the switch increments the appropriate failure count.

Table 14: Commands for Clearing or Displaying Dynamic ARP Inspection Statistics

Command	Description
clear ip arp inspection statistics	Clears dynamic ARP inspection statistics.
show ip arp inspection statistics bridge-domain id	Displays statistics for forwarded, dropped, MAC validation failure, IP validation failure, ACL permitted and denied,and DHCP permitted and denied packets for the specified bridge domain. If no bridge-domain is specified, the router displays information only for bridge domains with dynamic ARP inspection enabled (active).

Clearing or Displaying Dynamic ARP Inspection Logging Information

To clear or display dynamic ARP inspection logging information, use the privileged EXEC commands in table below:

Table 15: Commands for Clearing or Displaying Dynamic ARP Inspection Logging Information

Command	Description
clear ip arp inspection log	Clears the dynamic ARP inspection log buffer.
show ip arp inspection log	Displays the configuration and contents of the dynamic ARP inspection log buffer.

Additional References

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

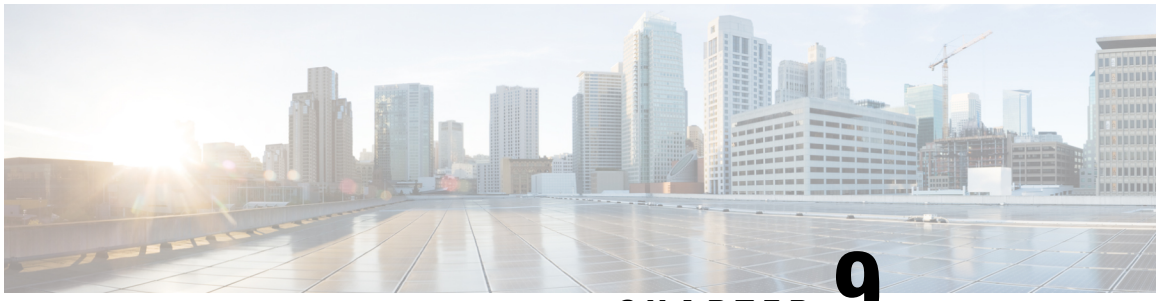
Feature Information for Configuring Dynamic ARP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 16: Feature Information for Dynamic ARP

Feature Name	Releases	Feature Configuration Information
Configuring Dynamic ARP	Cisco IOS XE Release 3.13.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D).



CHAPTER 9

IP Source Guard for an Interface

An IP source guard filters a source IP address on a layer 2 port and prevents malicious hosts from impersonating a legitimate host. The feature uses dynamic DHCP snooping and static IP source binding to match IP addresses to hosts on untrusted layer 2 access ports.

Initially, all IP traffic on the bridge domain associated with the DHCP snooping in a particular interface is blocked except for DHCP packets that are captured by DHCP snooping. After a client receives an IP address from the DHCP server, or after static IP source binding is configured by the administrator, the IP source guard feature automatically creates TCAM entries to permit that traffic. Traffic from other hosts is denied. This filtering limits the ability of a host to attack the network by claiming the IP address of a neighbor host.

Restrictions for IP Source Guard

- IP Source Guard (IPSG) configuration is supported only on interface level at 12 bridge domain interfaces.
- IPSG configuration works only if DHCP snooping binding is enabled.
- Only IP filtering is supported. IP MAC filter mode is not supported.
- IPSG configuration is not supported on port-channels, trunk EFP, and on BDI interfaces.
- IPSG is not supported on routed interfaces, layer2 and layer3 VPN and VRF.
- IPSG is supported only on video template.
- The IPSG entries are in IPv4 Tunnel TCAM region of ASIC. Since this a sharing model, any feature contributing more entries in this region impacts the scalability of other features.
- IPv6 is not supported. IPv4 and IPv6 packets that have IPv6 as first header is included under this restriction.

- Due to IPv4 tunnel TCAM region space limitation, only 1000 TCAM entries are supported. So, only 1000 IPSG entries are supported (including permit and deny entry). This impacts only the IP packets. Layer2 packets flow is not affected.
- If PBR and IPSG are enabled in a node at the same time, 1000 entries are shared by PBR and IPSG based on first come, first serve basis. If PBR is not enabled in the node, IPSG can be scaled to 1000.
- As IPSG and PBR share the same region, for a particular interface, these features are mutually exclusive.
- PBR is not supported on BDI that is associated with the IPSG enabled interface.
- ACL on EFP and IPSG perform the same functionality, that is, to deny or permit traffic on the EFP. So, when both of these features are enabled in the same EFP, both lookups are launched in parallel. So, either of the features deny the non-matching traffic. When ACL gives permit action and IPSG gives deny action for the same traffic, packets get denied, and vice versa. Only when both the features give permit action, traffic is permitted. Hence, though there is no restriction for configuring both the features on the same EVC, ideally these two features should be considered mutually exclusive.

Configuring IP Source Guard

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot/port* or **interface tengigabitethernet** *slot/port* or **interface port-channel** *type*
4. **ip verify source bridge-domain dhcp-snooping**
5. [**no**] **service instance** id **ethernet** [service-name]
6. **encapsulation dot1q** *vlan-id*
7. **rewrite ingress tag** {**push** {**dot1q** *vlan-id* | **dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **pop** {**1** | **2**} | **translate** {**1-to-1** {**dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **2-to-1 dot1q** *vlan-id* | **dot1ad** *vlan-id}*| 1-to-2 {dot1q *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **2-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id*}} **symmetric**
8. [**no**] **bridge-domain** *bridge-id*
9. **exit**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface gigabitethernet** *slot/port* or **interface tengigabitethernet** *slot/port* or **interface port-channel** *type* **Example:** `Switch(config)# interface gigabitethernet 0/1`	Specifies the interface to configure. • *slot/port* - Specifies the location of the interface. • *type* - Specifies the port channel interface.
Step 4	**ip verify source bridge-domain dhcp-snooping** **Example:** `Switch(config-if)# ip verify source bridge-domain dhcp-snooping`	Enables the IP source guard states. The **dhcp-snooping** option applies the feature to all VLANs on the interface that have DHCP snooping enabled.
Step 5	[**no**] **service instance** id **ethernet** [service-name] **Example:** `Switch(config-if)# service instance 101 ethernet`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**encapsulation dot1q** *vlan-id* **Example:** `Switch(config-if-srv)# encapsulation dot1q 13`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
Step 7	**rewrite ingress tag** {**push** {**dot1q** *vlan-id* | **dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **pop** {**1** | **2**} | **translate** {**1-to-1** {**dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **2-to-1 dot1q** *vlan-id* | **dot1ad** *vlan-id}*| 1-to-2 {dot1q *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **2-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id*}} **symmetric** **Example:** `Switch(config-if-srv)# rewrite ingress tag pop 1 symmetric`	Specifies the tag manipulation that is to be performed on the frame ingress to the service instance. **Note** In order for the device to distinguish if the packet is DHCP, all tags must be in pop state ; push and translate states are not supported.
Step 8	[**no**] **bridge-domain** *bridge-id* **Example:** `Switch(config-if-srv)# bridge-domain 12`	Binds the service instance to a bridge domain instance where *bridge-id* is the identifier for the bridge domain instance.
Step 9	**exit** **Example:** `Switch(config-if-srv)# exit`	Exits service instance configuration mode.
Step 10	**end** **Example:** `Switch(config)# end`	Exits configuration mode.

Configuring IP Source Guard With Static IP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip source binding** *mac-address* **bridge-domain** *bridge-id* **interface** *type mod /port*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**ip source binding** *mac-address* **bridge-domain** *bridge-id* **interface** *type mod /port* **Example:** `Switch(config)# ip source binding 000A.000A.000A bridge-domain 52 1.1.1.163 interface Gi0/0/0`	Adds the static entry.
Step 4	**end** **Example:** `Switch(config)# end`	Exits configuration mode.

Example

This example shows how to configure IP source guard:

```
Switch# enable
Switch# configure terminal
Switch(config)# interface Gigabitethernet0/0/1
Switch(config-if)# service instance 71 ethernet
Switch(config-if-srv)# encapsulation dot1q 71
Switch(config-if-srv)# rewrite ingress tag pop 1 symmetric
Switch(config-if-srv)# bridge-domain 10
Switch(config-if-srv)#exit
Switch(config-if)# ip verify source bridge-domain dhcp-snooping
```

Verification

Use the **show ip verify source** to verify the configuration:

```
router# show ip verify source
Interface  Filter-type  Filter-mode  IP-address       Mac-address        BD
---------  -----------  -----------  ---------------  -----------------  ----------
Gi0/0/5    ip           active       10.0.0.2                            10
Gi0/0/5    ip           active       10.0.0.3                            10
Gi0/0/5    ip           active       10.0.0.4                            10
Gi0/0/5    ip           active       10.0.0.5                            10
Gi0/0/5    ip           active       10.0.0.6                            10
Gi0/0/5    ip           active       10.0.0.7                            10
Gi0/0/5    ip           active       10.0.0.8                            10
```

Displaying IP Source Guard Information

To display IP Source Guard PACL information for all interfaces, perform this task:

Command	Purpose
Router# **show ip verify source** [**interface** *interface-name*]	Displays IP Source Guard PACL information for all interfaces on a switch or for a specified interface.

```
Router# show ip verify source interface gig0/0/1
Interface  Filter-type  Filter-mode  IP-address       Mac-address        Vlan
---------  -----------  -----------  ---------------  -----------------  ----
Gi0/0/7    ip           active       deny-all                            10
```

Displaying IP Source Binding Information

To display all IP source bindings on all interfaces, perform this task:

Command	Purpose
Router#show ip source binding [dhcp-snooping]	Displays all IP source bindings. The dhcp-snooping filter displays all VLANs on the interface that have DHCP snooping enabled.

This example shows how to display all IP source bindings configured on all the interfaces:

```
Router#show ip source binding
MacAddress          IpAddress        Lease(sec)  Type           BD    Interface
------------------  ---------------  ----------  -------------  ----  --------------------
04:62:73:2B:2A:3F   10.0.0.32        105         dhcp-snooping   10
GigabitEthernet0/0/4+Efp1 tag 10
Total number of bindings: 1
```

This example shows how to display only dynamic IP source bindings configured on all the interfaces:

```
Router#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           BD    Interface
```

```
------------------  ---------------  ----------  -------------  ----  ---------------------
00:00:00:4A:67:F4   30.0.0.188       44          dhcp-snooping   30
GigabitEthernet0/0/5+Efp3 tag 30
00:00:00:4A:65:8B   10.0.0.173       14          dhcp-snooping   10
GigabitEthernet0/0/5+Efp1 tag 10
00:00:00:4A:66:4B   20.0.0.184       51          dhcp-snooping   20
GigabitEthernet0/0/5+Efp2 tag 20
00:00:00:4A:66:D6   20.0.0.125       47          dhcp-snooping   20
GigabitEthernet0/0/5+Efp2 tag 20
00:00:00:4A:67:37   30.0.0.199       37          dhcp-snooping   30
GigabitEthernet0/0/5+Efp3 tag 30
```

Troubleshooting

Troubleshooting scenarios for IP Source Guard feature:

Problem	Solution
IP source guard not enabled	Use **show ip verify source** command to check if the entries exist.
DHCP snooping failures	1. Verify whether or not the issues are specific to DHCP snooping or IP source guard. Use the show **ip dhcp snooping binding** command to check the DHCP snooping bindings. If the expected entry is missing, debug the DHCP snooping sessions and share the output with TAC. 2. If the entry is displayed then check IP source guard configuration on the interface. If the issue persists, contact TAC.